MATHEMATICAL AND COMPUTATIONAL DEVELOPMENTS FOR BAYESIAN INFERENCE OF DAMAGE IN STRUCTURAL COMPONENTS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Andrew Emanuel Loeb

August 2017

MATHEMATICAL AND COMPUTATIONAL DEVELOPMENTS FOR BAYESIAN INFERENCE OF DAMAGE IN STRUCTURAL COMPONENTS

Andrew Emanuel Loeb, Ph.D.

Cornell University 2017

Non-destructive evaluation of structural components is critical for reducing costs from unnecessary replacements and maintenance. We study the utility of a non-contact modality for the inspection of structural components for the detection and characterization of damage in the form of through cracks and localized corrosion. We focus on the characterization of very small damage with a thermal imaging technique, since sensitivity to early stages of deterioration allows for simpler and less expensive repair than if a flaw propagates and becomes more threatening. The damage we consider interacts with the flow of heat so that a structure's thermal response to a known energy input can provide useful information for inference. Strategies are developed for optimizing a noise-sensitive thermographic experiment to produce optimal data for determining the otherwise hidden properties of the structure. Bayesian inference methods are developed for these tasks, as well as a novel heterogeneous computing method for rapidly simulating the conduction of heat through a three dimensional structure having heterogeneous material properties.

Our optimized experiment design for crack characterization is found to produce the same quality of inference as previous settings with much more expensive equipment (e.g. powerful lasers and sensitive IR cameras). It is also found that detection and inference can be done on corrosion pits only millimeters deep in the rear side of a steel panel using thermal observations from the front side.

BIOGRAPHICAL SKETCH

Andrew Emanuel Loeb was born in Longview, WA, USA on 21 December 1990. His parents emphasized the importance of a good education from the start. This instilled an attitude of excellence in Andrew, not just towards academics but towards all his endeavors.

While Andrew's proudest achievements are academic in nature, at Kelso High School Andrew also competed in soccer, track, and knowledge bowl. As captain of the knowledge bowl team, he led the team to the state championship tournament. During his summers, he worked on his grandfather's farm, with whom he developed a close relationship. His grandfather imparted upon Andrew an improved work ethic and Andrew's oft-quoted saying "There's nothing like having the right tool for the job." Andrew's work ethic benefited him greatly during the school year, as despite participating in athletics and other extracurriculars, Andrew excelled in every advanced placement course his school offered, particularly enjoying chemistry and literature.

With a stellar high school record and a scholarship awaiting him, Andrew chose to attend Harvey Mudd College, a college known for its strong technical program, an emphasis on the humanities, and close working relationships between professors and students. In the first semester of his sophomore year, Andrew declared as an engineering major. In his second semester as a sophomore, he took an engineering mathematics course taught by Professor Lori Bassman, which he immensely enjoyed. In this course, Andrew started to develop a close working relationship with Professor Bassman and a revitalized interest in mathematics. With a recommendation from his image processing professor, Professor Bassman offered Andrew a position in the Laspa Fellowship in Applied Mechanics, where he would work on microscopy data segmentation.

That summer, Andrew traveled to Sydney, Australia to work with Professor Bassman and other members of the tightly-knit Laspa Fellowship. In an extremely collaborative and ego-free research group, Andrew learned how to ask the right questions and honed his analytical skills. By the fall, Andrew was leading the Laspa Fellowship. Returning to Harvey Mudd in the fall of his senior year, Andrew decided to direct his focus on computational engineering. His choice was further reinforced by his experience of leading his clinic team in applying machine learning techniques to control of prosthetic arms.

While Andrew's academic credentials were without question, so was his character. Andrew directed the Dean of Students high-school tutoring program for three years. The Director of Learning Programs, personally requested Andrew to become the only non-math major tutor for the core math curriculum. Andrew also tutored for the Academic Excellence program and Tau Beta Pi. In addition to his passion for teaching, Andrew was also dormitory president for two years, a member of SWE, and an active member in the sailing club. With outstanding credentials, Andrew was selected for the National Science Foundation Graduate Research Fellowships Program and chose to attend the Applied Mathematics program at Cornell University.

At Cornell, Andrew focused first on becoming a better-rounded mathematician by taking extra classes outside of the graduate program. During his first year, he met Professor Christopher Earls, with whom he quickly developed a great relationship. Professor Earls quickly earned Andrew's respect for his leadership style, humility, and trust in his students. In particular, Chris trusted Andrew to learn about heterogeneous computing, which would have high initial research costs but sustained benefits. Their relationship has not only led to three research projects, but also to a friendship outside of their research.

For my parents and grandparents.

ACKNOWLEDGEMENTS

I wish to thank my advisory committee for their guidance and support through the Center for Applied Mathematics. In particular, a special thanks to Chris Earls for teaching through his examples of hard work, trust, and thoughtfulness. He showed me both courage and humility that serve the pursuit of success within an interdisciplinary career. If I didn't have Chris, I could not have built one.

My research has been financially supported by many sources, including the National Science Foundation (DGE-1144153), Cornell University, several undergraduate scholarships within and outside Harvey Mudd College, as well as my parents and grandparent. I am very grateful for everything that helped me get this far.

Lori Bassman was pivotal in my academic career. She began my work in advanced research, helped me to get into graduate school, and taught many of the research skills that I used along the way. I would also like to thank several of my peers that served in a mentorship role for me: Hufsa Ahmad, David Golay, and Hyung Joo Park.

I thank my friends and family for every part they played in helping me.

Lastly, I deeply thank Xanda Schofield for her support, friendship, and love.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Dedication . . . v
Acknowledgements . . . vi
Table of Contents . . . vii
List of Tables . . . ix
List of Figures . . . x

1 Introduction **1**

2 Thermographic Characterization of Sub-Pixel Sized Through Cracks **5**
2.1 Introduction . . . 5
2.1.1 Scope and Organization . . . 6
2.2 Background and Motivation . . . 6
2.3 Problem Description . . . 9
2.3.1 Analytical Solution . . . 12
2.3.2 Optimization Hypothesis . . . 15
2.3.3 Forward Modeling . . . 17
2.3.4 Inverse Problem Formulation . . . 19
2.3.5 Practical Considerations . . . 20
2.4 Results and Discussion . . . 22
2.4.1 Validation of Optimization Hypothesis . . . 22
2.4.2 Optimal Trends for Parameters . . . 24
2.4.3 Optimal Laser Position for a Bounded Domain . . . 26
2.4.4 MCMC-Based Inversion Results Using Optimal Experimental Conditions . . . 27
2.5 Conclusions . . . 32

3 Heterogeneous Computing Methods for Simulating Heat Conduction in Heterogeneous Materials **34**
3.1 Introduction . . . 34
3.1.1 Scope and Organization . . . 35
3.2 Background and Motivation . . . 35
3.3 Problem description . . . 38
3.3.1 FE Formulation I (PDE) . . . 38
3.3.2 FE Formulation II (Assembly-Free Methods) . . . 40
3.3.3 Preconditioned Conjugate Gradient . . . 41
3.3.4 OpenCL Heterogeneous Computing Framework . . . 44
3.4 Implementation of Assembly-Free Methods . . . 48
3.4.1 Mesh Geometry . . . 48
3.4.2 Previous Strategies . . . 49
3.4.3 Implementation 3: FG DbD with Memory Coalescing . . . 53
3.4.4 FG DbD with Memory Coalescing on multiple GPUs . . . 57

3.5 Experiments and Discussion . . . 60
3.5.1 Performance Comparison . . . 60
3.5.2 Multiple GPUs . . . 62
3.5.3 Further CPU Comparison . . . 64
3.5.4 3D Coefficient Inverse Problem . . . 65
3.6 Conclusions . . . 70

4 Thermographic Detection and Characterization of Pitting Corrosion in Structural Components 72
4.1 Introduction . . . 72
4.1.1 Scope and Organization . . . 73
4.2 Background and Motivation . . . 73
4.3 Problem Description . . . 77
4.3.1 Mathematical Formulation . . . 77
4.3.2 Forward Modelling . . . 81
4.3.3 Practical Considerations . . . 83
4.4 Bayesian Infernce Methodology . . . 85
4.4.1 Bayesian Inference Background . . . 85
4.4.2 Gaussian Process Background . . . 88
4.4.3 Detection Procedure . . . 90
4.4.4 Bayesian Experiment Optimization . . . 94
4.4.5 Detailed Characterization: Markov Chain Monte Carlo . . . 99
4.5 Results and Discussion . . . 101
4.6 Conclusions . . . 104

A Chapter 1 of Appendix 107
A.1 PDE Solution . . . 107
A.2 MCMC Details . . . 109
A.3 Convergence Analysis . . . 114

B Chapter 2 of Appendix 116
B.1 GPU PCG Algorithm Details . . . 116
B.1.1 Memory . . . 116
B.1.2 Kernels . . . 118
B.1.3 PCG Again . . . 125

Bibliography 128

LIST OF TABLES

2.1 Optimal trends for parameters in the crack characterization problem. 24
2.2 Aggregated mean and variance from two sets of 20 MCMC samplings. 30

3.1 Ratio of the rate of increase of wall clock time with increasing degrees of freedom between the sparse matrix CPU method and the linear asymptote of each GPU method. 61
3.2 Full performance comparison between the sparse matrix CPU method with incomplete Cholesky preconditioning and our best performing implementation. 64

4.1 Notation that is used throughout this work. Within the corrosion detection laser scan, some values are time-dependent. 86
4.2 Six corrosion scenarios and the error in estimating their location after the detection stage and Bayesian optimization stage. 102
4.3 Posterior mean and variance MCMC sampling. 103
4.4 Posterior statistics from MCMC sampling. 103
4.5 Posterior mean and variance for cross-sectional area 103
4.6 Corrosion pit location error after each stage of analysis. 104

B.1 Memory buffer types and initialization values. If a number is given for initialization, the specified number of bytes is allocated. 118

LIST OF FIGURES

2.1 (left) Inspection schematic annotated with relevant domains for the mathematical idealization Equation (1). (right) Close-up view of the local flaw neighborhood with overlaid pixel boundaries from the IR imaging system. The crack location (center), major axis, and minor axis are marked. 12

2.2 (left) 2D cross section of the radially symmetric analytical solution Equation (2) and (right) its gradient magnitude at varying durations of the laser pulse. 13

2.3 Infinite array of images of a square domain with an example of the symmetric positions for a laser peak in each image. 14

2.4 2D cross sections of analytical solutions at varying distances to domain boundary, marked on the x axis, with and without the use of images to satisfy the insulated boundary condition. 14

2.5 2D cross section of simulation response with varying laser peak positions, $\vec{x}_l = (0.22, 0), (1.57, 0)$, and $(6.5, 0)$ for a crack location $\vec{x}_c = (0, 0)$ with major axis in the y direction, showing a larger jump in temperature when the laser is positioned so that $\vec{x}_c$ coincides with the location of maximal gradient of the temperature response. 16

2.6 (left) Likelihood PDF of additive noise η for various ratios of NETD/ΔT with NETD$= 1$. (right) Standard deviation of such PDF asymptotically approaching 1. 22

2.7 (left) 2D cross section of the gradient magnitude function emphasized at seven laser offset distances. (right) Scatter plot of average MCMC posterior variance vs. analytical gradient for samplings performed using these distances. 23

2.8 Maximal gradient magnitude as a function of pulse duration for an optimally-located laser spot as an example of asymptotic behavior. The dashed line denotes the limiting value. 26

2.9 Vector diagram showing optimal laser location given a crack location over the 100 mm $\times$ 100 mm domain. The arrow bases are independently considered crack locations, with each arrow pointing at the optimal location for the laser peak to characterize a crack at its base. A circle denotes that a crack in its center is optimally characterized using a laser peak anywhere along its circumference: that is, asymmetry in the thermal gradient is negligible. 28

2.10 (upper) Marginal posterior histograms for one of the twenty MCMC samplings in set 1, and (lower) set 2. Solid vertical lines denote the means of the samples, while dashed lines denote the true values used to produce surrogate inspection data. 31

2.11 Joint posterior samples gathered from MCMC samplings used in the convergence study. 32

3.1 30×30×10 mm domain with tetrahedral meshing and three functions describing changes in material properties. A boundary between two materials can be abrupt (left) or smoothed over many elements (middle). Heat conduction can also be simulated on a domain with more complex dependence between material properties and spatial location (right) with negligible computational burden, provided that the properties have a closed functional form (here, the shading is computed as $x^2 - 0.2y^2 + 10z$ at each vertex and the values averaged over each element). 42
3.2 Emphasized view of the tetrahedral subdivisions of each small cube in the mesh. Each finite element is characterized by four vertices of the cube. 49
3.3 (left) Local data requirements for the first assembly-free MVM method. Elemental assembly matrices are required for six tetrahedral elements that comprise a cube in the mesh, as well as entries of the input vector corresponding to its eight corners. One element is emphasized and isolated (right), denoting the responsibilities of a single work item. There are 24 work items, within each work group for all such finite element-DoF pairs, required to assemble the output vector. 51
3.4 Necessary local data and work item responsibilities for the second assembly-free MVM method. (left) Data that is loaded from adjacent locations in memory for the input vector are connected by a green line to emphasize the potential for coalesced memory loading. Note that this figure is truncated for clarity and that the method actually loads 64 consecutive entries to local memory. (right) A single work item requires more data, but fully computes an entry of the output vector, denoted by a filled point. As before, the data required by a single representative work item is emphasized from local memory on the left. 54
3.5 Necessary local data and work item responsibilities for the third assembly-free MVM method. (left) Four coalesced reads from global memory provide all of the necessary input vector data. Note that this figure is truncated for clarity and that the method actually reads 32 consecutive entries to local memory with each coalesced memory read. (right) The representative work item computes contributions for all degrees of freedom associated with its finite element. 55

3.6 Memory access and computational partitioning pattern for the coalesced DbD MVM method. The orange nodes correspond to data in global memory, while the green nodes represent local memory in each work group. The short gray lines within each work group represent the tetrahedral elements which must have access to the vertex data at all four corners to compute their assembly contributions. Four such coalesced reads from global memory are performed to provide the tetrahedra with their necessary data. 56
3.7 Element padding for a 3×3×n mesh as seen from a top-down perspective. All of the data is loaded, but only contributions from solid elements are stored. This allows coalesced access to global memory to be used without interruption, at the cost of the extra discarded computation. 57
3.8 (left) splitting of vertices between two devices according to a user-specified fraction m. (right) Data that must be transfered between devices at each iteration. 59
3.9 (left and center) Time per PCG iteration for four GPU implementations and a serial sparse implementation of the same algorithm. Results are computed for each method as far as hardware limitations would permit. (right) Total number of iterations required to solve a transient heat conduction boundary value problem with 50 time steps. 61
3.10 Time per PCG iteration for four single-precision GPU implementations and the (double-precision) serial sparse implementation of the same algorithm for comparison. Results are computed for each method as far as hardware limitations would permit. . . . 63
3.11 Parameterized corrosion pattern within a steel gusset plate. The parabolic corrosion boundary is "anchored" at the points shown in black, and grows out away from the rear boundary. Heat is input to the system on the front face, where the resulting temperature profile is also recorded. 67
3.12 Algorithmic flowchart for solving a coefficient inverse problem with MCMC and the heterogeneous computing FEM methods described in this work. 69
3.13 (left) Trajectory of the 2500 sample Markov chain. (right) The same samples plotted as a histogram which approximates the distribution $p(\theta|\mathcal{D})$. 69

4.1 Diagram of the problem domain, labeled according to the mathematical formulation. (left) A 3D steel panel under inspection, with (right) a 2D cross-section into its depth through the center of a pit of corrosion. 79

4.2 Inference pipeline of the proposed process. Dashed blue lines denote IR measurement data (simulated or experimental), while solid orange lines indicate learned data regarding the corrosion pit parameterization. 85
4.3 Corrosion detection procedure partly through a scan. (a) Simulated pixelated IR response to laser scan and (b) $\hat{Y}^{(85)}$. (c) The cumulative sum of deviations after 1.7 seconds and trailing the laser scan, denoted by the vertical black line. Two pixels are emphasized, with trajectories of their neighborhood cumulative deviations. (f) The negative log likelihood that the measurements from each pixel and its neighbors are the result of a Gaussian random walk. 91
4.4 Output from a corrosion detection scan after an anomaly is flagged. (a) The likelihood map showing a bright region that has exhibited an abnormal and spatially correlated thermal response, and (b) the cumulative deviation trajectory of the flagged pixel. (c) A 3D view of the likelihood map to compare with (d) the Gaussian surface fit that best estimates it. 93
4.5 Pointwise evaluations of the experiment optimization objective function $G(\vec{x}_l)$ for four different corrosion pit parameterizations (black dots). Gaussian process regression with hyperparameter learning via maximum marginal likelihood is performed for each set of data. The GP mean function and learned length scale a_2 are shown. Additionally, a black ring around each surface corresponds to a level set for an assumed noise of $\sigma_{\text{NETD}}\sqrt{n}$ away from the peak. 97
4.6 Five iterations of Bayesian optimization beginning from the Gaussian fit to the likelihood map in Figure 4.4. At each stage, the proposed location for $\vec{x}_l^{(k)}$ is used in an experiment, and the resulting value of $G(\vec{x}_l^{(k)})$ is used to update the GP fit. Countours of the GP mean, conditioned on observed data, are shown on top. The location that produced the greatest thermal signal, $\vec{x}_l^{+}$ at each iteration is highlighted in magenta, and the true value of the corrosion pit center $\vec{x}_c$ is shown in black. The expected improvement function at each iteration is shown below. 98
4.7 Results of Bayesian optimization after six iterations for five additional scenarios with the corrosion pit location and shape set according to Table 4.2. As in Figure 4.6, the black star denotes $\vec{x}_c$, and the laser spot point $\vec{x}_l^{+}$ which gave the greatest thermal response is highlighted in magenta. 102

4.8 Trial 1 histograms of MCMC samples estimating the marginal posterior distributions of each corrosion pit parameter. The true values for each parameter are denoted by a dashed blue line, while the posterior mean is shown in black. 105
4.9 Trial 1 joint posterior distributions for two pairs of corrosion pit parameters. (left) The location parameters do not exhibit strong correlation, and (right) the shape parameters do. Contour lines for the total corrosion volume function are overlaid in white. For both plots, the true values from the experimental corrosion pit are shown with blue dots. 106

A.1 Surface plot of the nondimensionalized response as a function of $\mathcal{X}$ and $\mathcal{T}$. 109
A.2 (left) Experimental lag 1 autocorrelation as a function of L for the for uncracked problem. (right) Experimental acceptance probability of the same data. 113
A.3 Trajectories of the $(1-\alpha)$ convergence diagnostic for a representative pool of MCMC samplings. 115

CHAPTER 1

INTRODUCTION

Thermographic non-destructive testing (NDT) is a collection of inspection methods through which hidden properties of a structure are inferred by the way the structure responds to heat [56, 62]. In this work, we explore an active thermography experimental setup applied to two problems of engineering interest. The experiment involves inputting energy from a modestly powerful (10-100 watt) laser and measuring temperature over a visible region with an infrared imaging sensor [69]. The two modes of damage that we consider are small, sub-pixel sized cracks through the depth of a thin aluminum sheet, and pitting corrosion on the inaccessible back side of a steel structure.

Both of these tasks present *inverse problems*: making quantitative estimates of parameters describing damage, based on a system response to a known input. In this case, the measurements are contaminated with realistic sensor noise, as well as discretized in space, time, and temperature to simulate the image capture physics. Therefore, the ill-posed nature of the backwards heat equation implies that direct inference of the damage parameters from data is difficult [21]. We use finite element method (FEM) simulations as *forward models* to furnish idealized hypothetical responses of systems with known properties. From these, Bayesian inference techniques are used to estimate probability distributions over the damage parameters of interest, with stochasticity driven by the noise in measurements.

The Bayesian framework allows one further stage of analysis. If we are able to make probabilistic statements based on the output of an experiment, then it is natural to try to modify the experiment so that it will yield optimal data

for this purpose. Even with a fixed set of equipment to use, there are several design choices to be made. For example, where should an inspector aim the laser? How long should the specimen be heated? Should the heat be allowed to dissipate before making a measurement? We consider such experiment optimization problems so that damage can be detected in its nascent stages, or with less expensive tools, to inform better maintenance decisions.

This dissertation is a collection of three papers. Each chapter consists of autonomous units with an introduction and a conclusion. References for all chapters are collected at the end.

In Chapter 2, we consider the problem of characterizing nascent stage through cracks in thin aluminum sheets. The sheets are thin enough that a two dimensional approximation for heat diffusion is justified. With this simplifying approximation, the analytical solution of the heat equation over an undamaged panel is known, and can be easily manipulated. Furthermore, FEM simulations are straightforward with constructive solid geometry and standard software [44]. Meshes with tens of thousands of degrees of freedom are sufficient for numerical accuracy, as well as rapid performance. We take as a point of departure a crack detection technique from the literature, so that we begin with an approximation of crack location, with its size and shape unknown. It is proposed that the characterization experiment can be optimized by analytically maximizing the gradient of the idealized solution with respect to any adjustable parameter. This hypothesis is tested through many simulations, where an optimal experiment is defined as one that yields data which in turn give an inverse problem solution with suitable confidence bounds. All of the experimental parameters are considered, with emphasis on the optimal location of the

laser spot with respect to the location of the crack. The chapter concludes with Markov chain Monte Carlo (MCMC) solutions to the crack characterization inverse problem. We see similar results between surrogate experiments that have been optimized according to our methods, and experiments using values from previous work and a laser source that is ten times more powerful.

The second mode of structural damage that we consider is that of corrosion on hidden regions of a structure. In this case, the domain and damage are three dimensional. A mesh that can resolve small, curved corrosion profiles can easily comprise hundreds of thousands, or millions of degrees of freedom. The demands of solving many thousands of such simulations as part of an inverse solution motivated the exploration of nonconventional algorithms and computer architectures. Chapter 3 documents the results of this exploration. An algorithm for rapid successive simulations over a domain with parametrically varying material properties is developed for graphics processing unit (GPU) computing architecture. Three implementations are compared, with varying interpretations of the FEM assembly operator, and the most effective implementation is adapted for use on dual GPUs.

In Chapter 4, the GPU algorithm is deployed as the foundation of a corrosion detection and characterization framework. Parallels can be made with Chapter 2, in that the process begins with a coarse sweep of the structure under consideration in order to give estimated locations of hidden damage. Next, a strategy for optimizing a non-destructive evaluation experiment is developed, specifically for the location of the laser spot. The strategy uses modern Bayesian machine learning tools and the same collection of experimental equipment to give posterior information regarding the location of corrosion pits, as well as optimal

data for the purposes of an inverse solution. Finally, Bayesian inference of the probability distributions over corrosion pit parameters is carried out using the tools from Chapter 3. Demonstrations are given for several sizes, shapes, and locations of corrosion pits.

CHAPTER 2

THERMOGRAPHIC CHARACTERIZATION OF SUB-PIXEL SIZED THROUGH CRACKS

2.1 Introduction

The use of pulsed laser thermography for non-destructive evaluation has been incrementally developed over the past decade. The method has been shown to be viable for detecting through cracks in the thin aluminum panels that compose the outer skin of airframes, for example. Previous work in the literature has mostly considered the detection and characterization of cracks that are several millimeters in length [39, 40, 60, 9, 13]. Flaws of this size may have already developed past the point of safe operation. In the current paper we focus on nascent-stage defects: cracks that are contained within the area measured by a single pixel of the infrared imaging system which is used in the inspection.

Cracks this small require a careful approach to characterize with currently-available tools. In particular, we aim to design an inspection modality that uses no more than an inexpensive infrared imaging system (e.g. 10,000 USD) and laser of modest power (e.g. 10 W). The specifications of the thermal camera we model are taken from an entry-level research instrument, and the laser power we consider in our simulations is consistent with previous laboratory experiments in the literature [39, 40, 60, 32]. The development of the proposed inspection design methodology will be done using a rigorous consideration of the mathematical theory of heat conduction. Each choice in the setup and performance of the thermal measurement is made to provide optimal data for the characterization of a sub-pixel crack in light of the theory. With the general util-

ity offered by the proposed framework for optimized inspection design, flaws can be detected and characterized more clearly, and at earlier stages of their formation. This will result in more effective strategies for safe and inexpensive maintenance across a range of material properties.

2.1.1 Scope and Organization

This paper is divided into five sections. In Section 4.2, the history of active thermography, as it has been developed for this class of problem, is summarized. In Section 4.3, the mathematical framework is described in terms of a partial differential equation and its solution. The hypothesis on which our method is founded is motivated and stated, and details about the practical solution to both the forward and inverse problems are described in that same section. Section 4.5 begins with the validation of our optimization hypothesis with numerical simulations. Then the optimization framework is carried out for characterizing sub-pixel cracks in thin metallic panels. The results of a stochastic inverse solution based on simulated data are presented in Section 2.4.4. Finally, Section 4.6 summarizes the conclusions of this paper and the significance of our method.

2.2 Background and Motivation

Infrared (IR) thermography is a nondestructive, noncontact evaluation technique characterized by using the thermal signal emitted from an object's surface in order to infer its internal structure [56]. There are two broad subclasses of thermography, founded on either a passive or an active thermal signal. In

the case of active thermography, a heat source such as a flash lamp, continuous wave laser, or frequency modulated laser is used to impart heat energy into the specimen under evaluation. Characteristics of the heat source are known by the experimenter, as well as the salient thermal properties of the specimen. Then subsequent variations in the thermal response can be used to infer other unknown structural properties. A history of IR technology and the development of pulsed thermal NDT is presented in Reference [69].

Active themography is particularly well-suited for the detection of cracks penetrating through a thin, plate-like specimen. Preliminary studies demonstrated that, while cracks are among the most problematic types of defect for characterization with thermal NDT, they do block heat flow in the perpendicular direction [67, 36, 37, 53]. This means that the heat from the known thermal input will be obstructed and produce a thermal response significantly different from a similar domain without a crack. Further studies that approached this problem with all three branches of modern scientific understanding (mathematical theory, experiments, and numerical simulation) are also available in the literature [39, 40, 60, 9, 13].

Chatterjee et al. provided a comparison of three laser-driven modalities—namely, pulsed, lock-in, and frequency modulated techniques [13]. It was concluded that the pulsed mode provided the best signal-to-noise ratio for defects that penetrate up to 1 mm into a structure. We adapt this latter mode in our proposed inspection design scheme because the thin panels we consider here have a total depth of only 1 mm. Li et al. took steps to optimize the overall NDT process according to a mathematical model [39, 40]. An "optimum" laser offset distance from a crack was presented for scanning pulse laser-line and laser-

spot thermography, which was later used in experiments by others [60]. Finally, Burrows et al. performed experiments studying in-service testing conditions to determine the narrowest crack opening and lowest laser power required for crack detection with laser pulse thermography [9]. All of the foregoing studies showed promise when applied to specimens with defects of several millimeter length.

Other work has been done to characterize specific physical parameters of cracks that have been previously found or are assumed to exist. Schlichting et al. investigated the problem of determining the size of a surface crack that does not fully penetrate a thick structure with the precondition that its location was already known [60]. The depth and angle at which the crack penetrates the sample were then estimated through the use of pulsed laser thermography. More recently, Jeong et al. refined this problem to consider cracks of length less than a millimeter, fitting entirely within a single pixel of an IR image [32]. Earlier direct methods could not give suitable information regarding the characteristics of such small flaws. In later work, automated stochastic procedures were developed to solve the inverse problem associated with optically unresolvable cracks, and demonstrate the feasibility of an inspection method founded on such an approach [17].

We continue the foregoing line of inquiry [32, 17], aimed at the characterization of small cracks penetrating through plate-like components, as well as developing an optimization framework for designing a suitable inspection modality for a particular context. That is, we have determined guidelines for the setup of an active thermography inspection to yield optimally precise estimates of a crack's true physical characteristics. The inspection phase here is the second in

a two-step process, as follows. First, the location of the crack must be somewhat known, e.g. Li et al. describe a second derivative method for locating defects from an IR image [40]. Alternatively, Bryan details a flexible and computationally fast method for detecting small cracks based on the reciprocity gap functional [7]. In a single pass, Bryan's method can detect many cracks in the presence of realistic noise and quantization error. While it does not reliably determine the crack length, the method is shown to provide a good estimate of the location and orientation of elongated sub-pixel cracks. With estimates of the locations of one or many cracks, we make a second pass over the sample, using the same equipment, so as to provide detailed information about these flaws. Ideally, the total inspection time should be brief, so it is better to not introduce new tools during this second pass: a condition that the present work satisfies. The subsequently gathered data can then be analyzed offline, to inform action that might be required in order to maintain or replace the fielded specimen. It is the second imaging pass which is the focus of the current work. We will henceforth also assume the scenario of a single narrow elliptical crack with a known orientation of the major axis, and with the location of its center, $\vec{x}_c$, known to within one millimeter. We seek to optimize the second pass inspection setup to fully characterize the crack. Our results will be robust under use with any investigation of small defects involving pulsed laser thermography and generalizable for any metallic material, based on its thermal properties.

2.3 Problem Description

The flow of heat within a solid medium is governed by the heat equation, a parabolic partial differential equation (PDE) [19]. In its full form, the heat equa-

tion models heat transfer over time, due to conduction, convection, and radiation within a given spatial domain. It may describe spatial and temporal addition of heat within the domain, or on its surfaces. Alternatively, these surfaces to be insulated and provide no heat transfer across them.

The current work considers a flat, 1 mm thick, 100 mm $\times$ 100 mm aluminum 5052 panel. The relevant thermal properties for this material are its *density*, ρ (2680 kg/m^3), *specific heat*, C (880 J/kg K), and *thermal conductivity*, k (assumed to be constant at 138 W/m K). The surfaces of the specimen, including the internal boundaries forming a crack, are assumed to be insulating.

Thermal energy provided from a laser beam is directed onto the front face of the panel. The laser is assumed to deposit thermal energy in the form of a Gaussian profile, centered at a point $\vec{x}_l$ [3]

$$f(\vec{x}) = \frac{2P}{\omega^2} \exp\left(-\frac{2\left|\vec{x} - \vec{x}_l\right|^2}{\omega^2}\right),$$

where P is the *laser power* and ω is the *beam width* (the radius at which the intensity has dropped to $1/e^2$ of its peak value).

In addition to the assumptions stated previously, our mathematical model is simplified by assuming that the panel under inspection has perfect absorptivity (i.e. behaves as an ideal black body), so that all of the laser energy is converted to heat. We also neglect convection and radiation surface effects. These effects are found to be negligible over the short inspection times considered, as supported by analytical and numerical considerations, and in the literature [60]. Lastly, the domain is idealized as two-dimensional, doing away with the 1 mm thickness of the panel. This is realized in all following computations by multiplying the material density of aluminum by the neglected *thickness*, h and setting the *thermal diffusivity*, $\kappa = k/\rho h C$. The 2D assumption is supported by

the *Biot number* for our material and domain geometry being 1.8×10^{-4}, since any value less than 0.1 implies that temperature variation within the solid is negligible compared with boundary effects [33]. The foregoing simplification is further validated with comparison of 2D simulations with full 3D simulations that include all nonlinear heat transfer terms. The subsequent difference in the temperature responses between 2D and 3D models vanishes after passing through our model for the imaging system with finite spatial and temporal resolution, while computation time is vastly reduced. It is noted that our approach to solving an inverse problem for crack characterization requires many iterations of the simulated inspections, so these simplifying assumptions are a practical necessity.

The resulting PDE boundary value problem that describes this inspection scenario is

$$\begin{cases} \frac{\partial T(\vec{x},t)}{\partial t} - \kappa \nabla^2 T(\vec{x},t) = f(\vec{x},t)/\rho h C & \text{in } \Omega \times (0,\infty), \\ -\frac{k}{h}\frac{\partial T(\vec{x},t)}{\partial \vec{n}} = 0 & \text{in } \Gamma_{\text{crack}} \cup \Gamma, \\ T(\vec{x},t) = T_0 & \text{on } \Omega \times \{t=0\}, \end{cases} \tag{2.1}$$

where $\vec{n}$ denotes the outward normal vector to the domain and T_0 is the initial, constant temperature of 25 °C. The domain Ω is the 2D surface of the panel with boundary Γ around the outside of the panel and along the edges of the crack, as depicted in Figure 2.1.

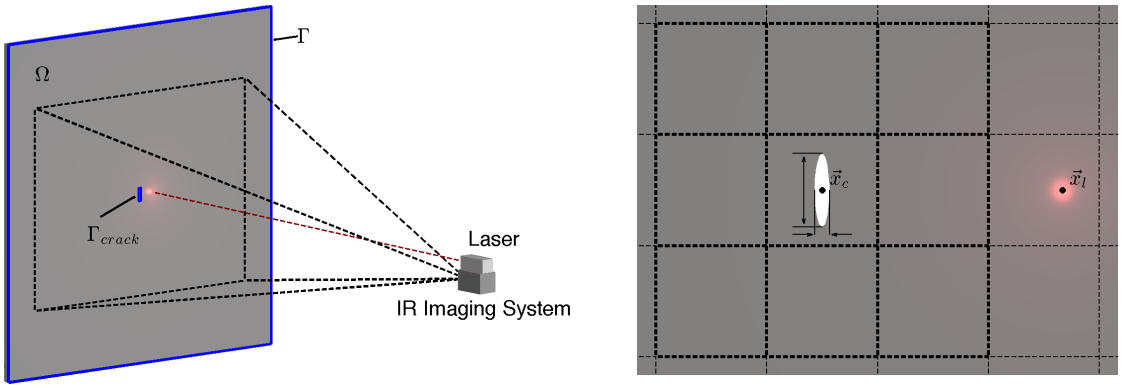

Figure 2.1: (left) Inspection schematic annotated with relevant domains for the mathematical idealization Equation (1). (right) Close-up view of the local flaw neighborhood with overlaid pixel boundaries from the IR imaging system. The crack location (center), major axis, and minor axis are marked.

2.3.1 Analytical Solution

The PDE boundary value problem above has a closed-form solution in the case that the domain is infinitely wide and uncracked ($\Omega = \mathbb{R}^2$). The analytic derivation of this solution is presented in A.1, along with a visualization of the time evolution of the general heat response.

The solution to Equation (4.1), but over the flawless, infinite domain is

$$T(\vec{x}, t) = \frac{P}{4\pi k}\left(\text{Ei}\left(-\frac{2\left|\vec{x}-\vec{x}_l\right|^2}{\omega^2}\right) - \text{Ei}\left(-\frac{2\left|\vec{x}-\vec{x}_l\right|^2}{\omega^2+8\kappa t}\right)\right), \tag{2.2}$$

where $\text{Ei}(z) = -\int_{-z}^{\infty} \frac{\exp(-t)}{t}\, dt$, the exponential integral. Note that the solution is radially symmetric around $\vec{x}_l$, the laser peak. A cross-sectional view of the temperature response and its gradient are shown in Figure 2.2.

To obtain the analytical solution to Equation (4.1) on a finite, but flawless, and insulated domain, the *method of images* [19] is used with Equation (2.2). By superimposing a replica of Equation (2.2), as if a second laser were positioned

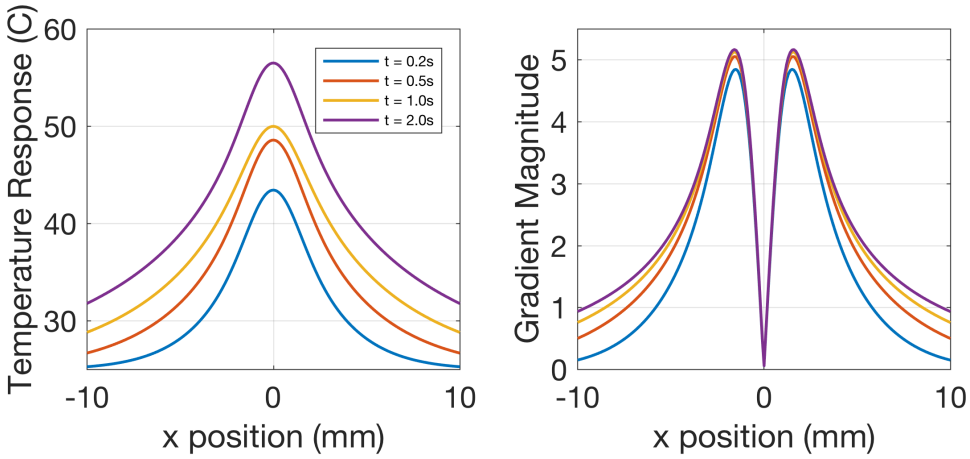

Figure 2.2: (left) 2D cross section of the radially symmetric analytical solution Equation (2) and (right) its gradient magnitude at varying durations of the laser pulse.

symmetrically across the domain boundary (a mirror image), the temperature response is symmetric across the boundary. This results in no net heat transfer: equivalent to an insulated boundary condition. For the 2D problem, an infinite array of shifted images of the domain are necessary for an arbitrarily precise solution, as demonstrated in Figure 2.3. While the boundary effects are negligible for a laser peak further than 20 mm from an insulated edge, they are profound for a laser aimed near the boundary. We found convergence with error smaller than the sensitivity of our imaging system model for all laser positions on a 100 mm $\times$ 100 mm domain. Nine images are required, which is feasible to implement computationally. This has all been further verified with comparisons to numerical simulations with the insulated boundary conditions imposed, using the same convergence criterion. Figure 2.4 compares this method with Equation (2.2) for varying laser peak positions.

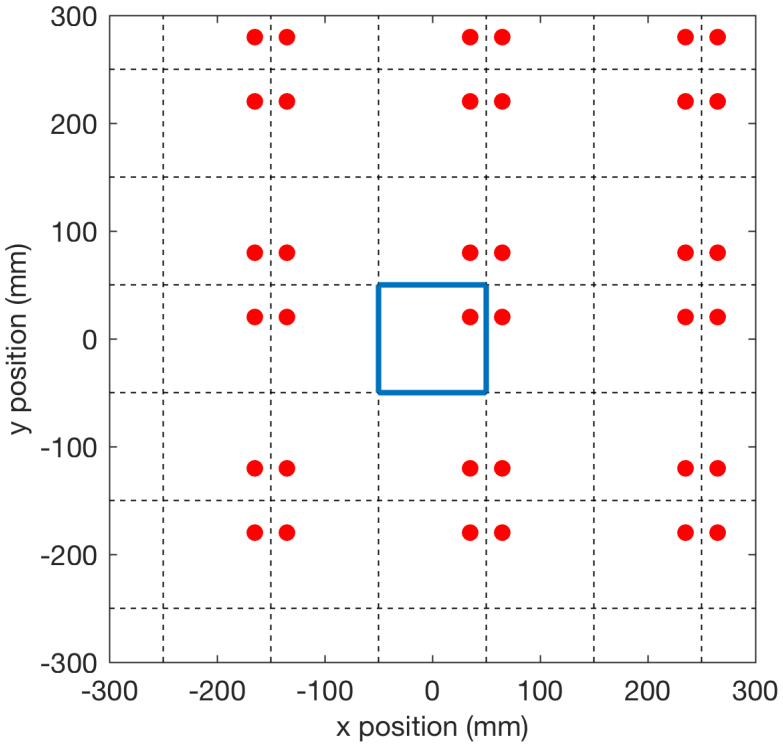

Figure 2.3: Infinite array of images of a square domain with an example of the symmetric positions for a laser peak in each image.

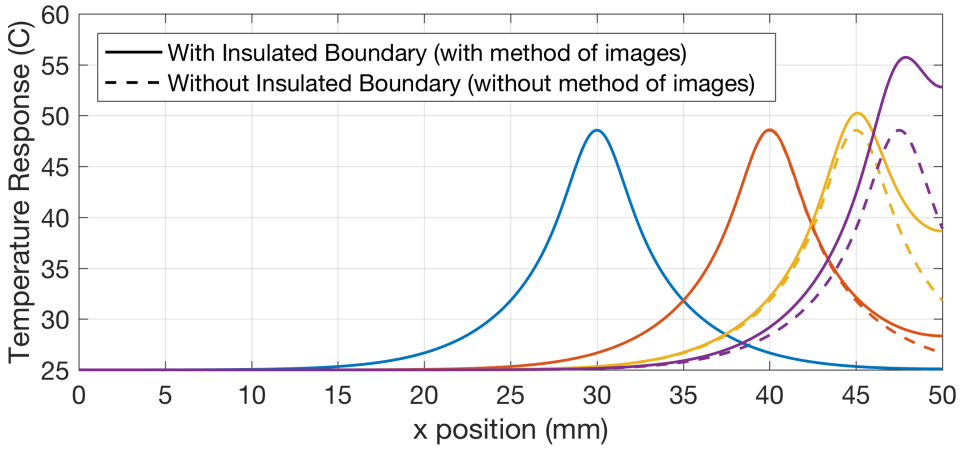

Figure 2.4: 2D cross sections of analytical solutions at varying distances to domain boundary, marked on the x axis, with and without the use of images to satisfy the insulated boundary condition.

2.3.2 Optimization Hypothesis

One goal of the current work is to determine the optimal inspection setup for characterizing small through cracks using pulsed laser thermography. This type of flaw interrupts the flow of heat through a thin panel domain, due to the crack's insulating effect, or viewed another way, its interruption in the continuity of the conducting medium. Hence, all of the available information about the crack is contained in the disruption it causes in the thermal response: that is, the difference between the measured thermal response in a cracked specimen and the expected response from an uncracked domain. This information is subject to both instrument noise and quantization error from the thermal imaging charge-coupled device (CCD), consisting in our case of assumed microbolometers. Therefore, the inspection data are more useful if the thermal disruption caused by the crack is large in the sense that it exceeds the measurement noise floor. A simple way to measure the strength of the disruption signal is by finding the thermal gradient from one side of the crack to the other in the recorded temperature response. A steep change in temperature is a sign that information about the crack is well-interpretable above the noise floor. Figure 2.5 shows the thermal response of a cracked domain caused by an identical laser pulse at three different values for the distances from the crack to the laser peak, $|\vec{x}_c - \vec{x}_l|$. A distance that is neither too close, nor too far, results in a high thermal gradient. By seeking to maximize this gradient, either simulation or trial inspections can be employed to manually tune parameters of interest in the design of a particular inspection protocol. This idea of maximizing the thermal gradient across the crack has been used by Li, et al. to provide a coarse estimate for the "optimal" distance between the laser peak position and a crack that is several millimeters long [40]. Furthermore, the selection of the optimal heating duration for a differ-

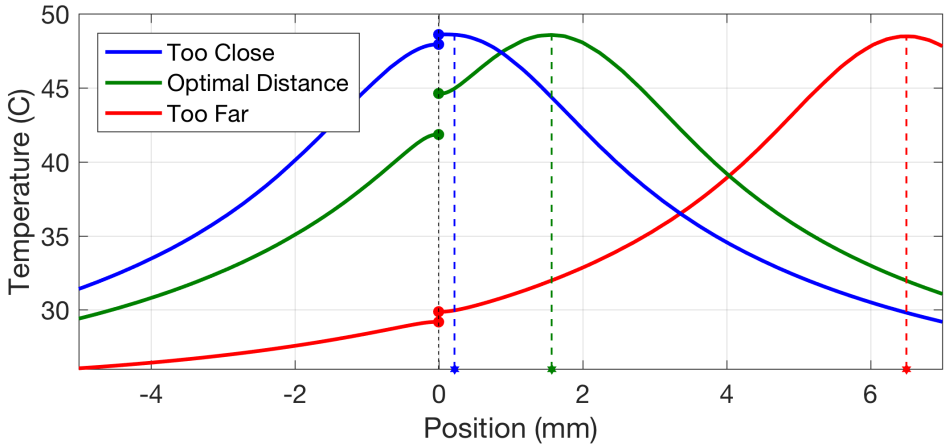

Figure 2.5: 2D cross section of simulation response with varying laser peak positions, $\vec{x}_l = (0.22, 0), (1.57, 0)$, and $(6.5, 0)$ for a crack location $\vec{x}_c = (0, 0)$ with major axis in the y direction, showing a larger jump in temperature when the laser is positioned so that $\vec{x}_c$ coincides with the location of maximal gradient of the temperature response.

ent problem in thermal NDT is discussed by Marinetti et al. [46]. These works use experimental data and numerical modeling, respectively, in the selection of their chosen inspection parameters.

The analysis in Section 2.3.1 gives a powerful tool for mathematically designing an optimal inspection setup. Since we have a closed-form expression for the thermal response in a flawless domain, we can analytically evaluate how changes in any inspection design parameter (e.g. material properties, laser power, etc.) affect the solution. In particular, we may find the spatial gradient vector field of Equation (2.2) itself, and take its scalar magnitude in order to obtain a function to maximize with respect to any inspection design parameter, or set of parameters. While this is not necessarily equivalent to maximizing the thermal gradient across a crack during inspection, there are no equivalent analytical results for the response in such a domain. We reconcile this issue with

the following hypothesis: *An inspection design parameter is optimized for crack detection and characterization when Equation (2.2) has maximal gradient with respect to that parameter.* This puts optimizing crack characterization in the framework of a constrained optimization problem that is not too difficult to solve. The validation of our hypothesis will be explored in Section 2.4.1.

2.3.3 Forward Modeling

The characterization of small cracks from thermal data can be posed as an *inverse problem*. One class of inverse problems are problems whose solution yields information about underlying parameters instantiating a system, leveraged against observations of that system's behavior. This is in contrast to a forward problem, which is the prediction of a system's behavior, based on its known physical properties and parameters. In our case, the forward problem is characterized by finding the thermal response of a flawed metal panel for a crack with known size, shape, and location. This can be solved using the heat equation and the *finite element* (FE) method, as a means of discretizing and numerically treating Equation (4.1).

In this work, we solve the forward problem as a weak form [19] using the FEniCS open source Python application programming interface (API) containing software classes that support the FE method [44]. The FEniCS API includes constructive solid geometry classes, with which it is straightforward to define a 2D domain with a crack of varying size and position. Using FEniCS, the heat equation PDE may then be solved over a graded, unstructured linear triangle FE mesh, which has a varying spatial resolution that is finer near the crack. A

spatio-temporal convergence study is carried out to find the appropriate spatial mesh refinement, as well as needed time step size. Convergence is confirmed across successive refinements as well as with comparison to the analytical solution and a full 3D simulation, using the criterion described above when considering discretization used in modeling the imaging system. Convergence is observed for time steps of 0.02 seconds and a spatial mesh of first-order Lagrangian triangles comprising approximately 33,000 nodes. The exact number of nodes depends on the size of the crack, as the mesh is generated after defining the geometry of the domain.

The finite element model furnishes an approximation to the heat response on a continuum. From this, realistic surrogate experimental data are generated, as if they were recorded by an actual thermal imaging system, through the following process. Field variable output from the FE mesh unstructured nodes are interpolated onto a rectangular grid, then integrated over the area which would be captured by each pixel of the hypothetical imaging system. Next, independent, identically distributed (i.i.d.) Gaussian measurement noise is added to each pixel reading, in a manner that is consistent with the noise-equivalent temperature difference (NETD) of our assumed microbolometer system. Finally, the data are rounded to be consistent with specified resolution, ΔT, that is associated with the quantization assumed in our image capture. We assume an entry level research camera (A325sc from FLIR Systems, Inc) as the basis for our modeled imaging system, which has a NETD of 50 mK, a standard temperature range from 0 °C to 350 °C, and 14-bit data representation, for a thermal resolution of $\Delta T = 0.02$ °C. The spatial resolution (320 $\times$ 240) and angle of view of this camera (6°) are determined to give a pixel size of 0.9 mm $\times 0.9$ mm with a standoff distance between the camera and the sample being tested of 2.7 m.

2.3.4 Inverse Problem Formulation

In this work, we use the *Markov chain Monte Carlo* (MCMC) method in solving the inverse problem of crack characterization [25]. This is a particular approach to solving inverse problems, but other methods exist. For a detailed explanation of the MCMC approach, as it applied to thermal imaging (e.g. crack position and size), see A.2. The advantage of MCMC is that it furnishes an inverse solution in the form of a *probability density function* (PDF), called the *posterior*, which reflects the relative probability for values of unknown parameters based on the considerations of both measured data and their measurement uncertainty (encoded into the *likelihood* PDF) along with prior beliefs regarding the model parameters (encoded in the *prior* PDF). Several statistics of interest can then be computed from these posteriors. The posterior *mean* and *variance* are particularly useful, since our focus here is optimizing the preceding inspection setup to provide informative data. We use the variance of a posterior distribution as a measure of precision within the data. An inspection design that results in a tight distribution of values (i.e. small variance) for a parameter is preferred over one giving a wide distribution. We may then compare *credible intervals* within the solution, using different inspection design parameters. That is, inspection parameters that yield high "confidence" in the inverse problem solution are favorable over those that succumb to noise, and fail to provide clear signals regarding the inversion parameters of interest.

2.3.5 Practical Considerations

We make a number of simplifying assumptions in the mathematical formulation of the crack characterization problem. Their justifications and limitations are enumerated here. First, it has been stated that this study is particularly concerned with through cracks in plate-like components, where the crack lengths are smaller than the imaging resolution. Although this assumption presents the primary challenge of the study, it is useful for our solution method. This is because our optimization hypothesis relies on finding specific values of geometric parameters that maximize the gradient of an analytic function. In particular, the crack location is described as a single point, $\vec{x}_c$, regardless of its actual size or shape. As a result, small flaws are better-represented in the proposed framework than larger cracks that may be easier to detect outright, in the first place. Another simplifying assumption adopted for the flawed panel is its absorbtivity of laser energy: we assume full absorbtivity, but this can be effectively relaxed, changing only the laser power coefficient to reflect the decreased energy that is transmitted into the panel.

Next, there are practical limitations on the parameterization of the laser that is used as the thermal energy source. A more powerful laser will produce higher contrast in a thermal image, but this cannot be relied on as the only means of detection enhancement for two important reasons. In addition to high cost and increased risk of personal injury, if a given laser is too powerful, it poses a risk to the sample under inspection. Earls [17] discusses sensitization, which may occur in aluminum 5052 above 260°C. To avoid damage to the specimen we are attempting to inspect, and to demonstrate that our proposed method can be used without prohibitively expensive hardware, we consider a 10 W laser.

The other parameter of an ideal Gaussian laser source is the spot size. Optical lenses can be used to manipulate the spot size of a laser with knowledge of the distance between the laser source and the target. To focus this spot size to be very small, the sensitivity to this distance becomes dramatic. A relatively high divergence angle must be used, so that small changes in the standoff distance result in large relative changes in spot size. For this reason, small spot sizes are not favorable from a practical perspective. Finally, it is noted that the Gaussian profile of laser energy, rather than, say, uniform over the spot size, is a well-supported assumption [55].

Lastly, we comment on the effect of quantization error resulting from the thermal camera data representation precision. Although i.i.d. Gaussian measurement error due to electronic sensor noise is captured in our model, and drives the stochasticity of the inverse problem solution, through the form of the likelihood PDF, the quantization error due to the bit depth of the A/D converter in our assumed CCD is not included directly in our modeled imaging system. Since we rely on sensitivity to slight changes in a temperature response, low thermal resolution is as deleterious to our aims as are contamination from electronic sensor noise. On the other hand, higher temperatures, and in particular the resulting larger gradients will reduce the relative importance of such quantization error. Thus the high-gradient optimization we present will overcome both forms of noise simultaneously. We investigate the tradeoffs between these two sources of error for varying severity, illustrated in Figure 2.6. For values of NETD/ΔT greater than 2, measurement noise is determined to dominate. This is the value at which the discretized normal distribution has a standard deviation within 1% of a smooth Gaussian. The camera we simulate here has parameters NETD=0.05 °C and ΔT=0.02 °C, giving a ratio of 2.5. Thus quantization

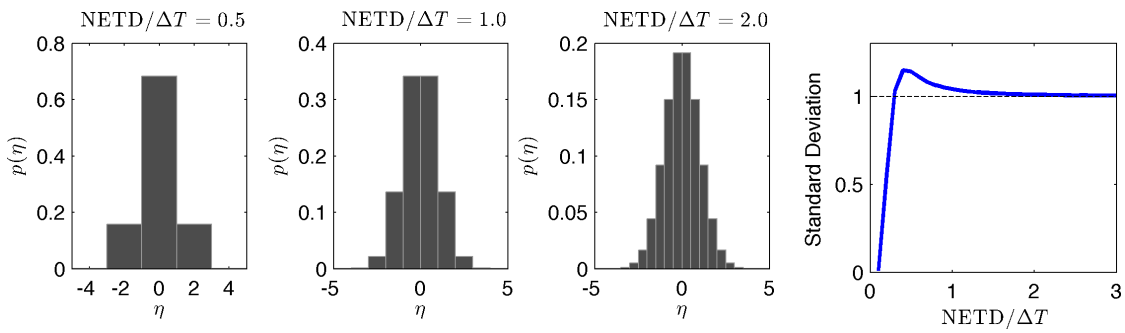

Figure 2.6: (left) Likelihood PDF of additive noise η for various ratios of NETD/ΔT with NETD$=1$. (right) Standard deviation of such PDF asymptotically approaching 1.

error may be neglected here.

2.4 Results and Discussion

2.4.1 Validation of Optimization Hypothesis

Before exploring the utility of the proposed optimized crack characterization inspection design, we first determine the validity of the hypothesis stated previously in Section 2.3.2. Towards this end, it is important to observe the relationship between the gradient magnitude of the analytic solution and the quality of inference in the associated inverse problem, prior to using this metric to optimize the inspection design. To do this, we isolate a single parameter, the distance between the crack and laser position, measured perpendicular to the crack major axis (the x direction), and perform several surrogate experimental MCMC runs. IR imaging system sensitivity is neglected in these trials, so that the only sources of randomness involved in creating estimates of posterior

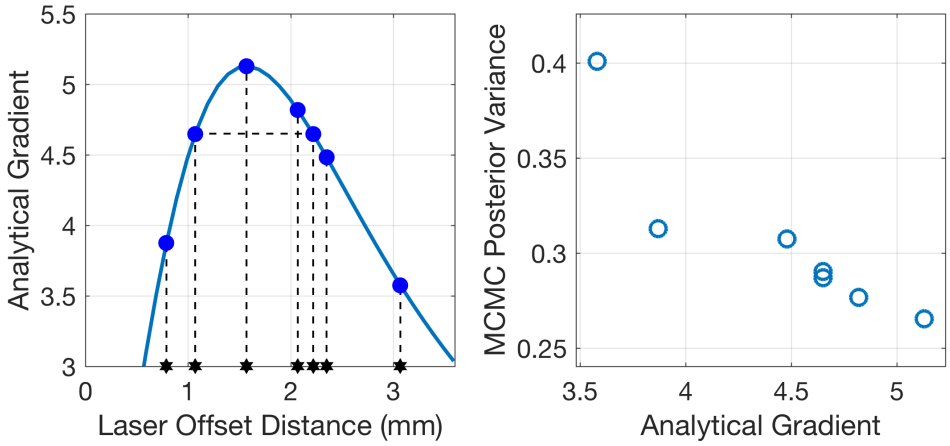

Figure 2.7: (left) 2D cross section of the gradient magnitude function emphasized at seven laser offset distances. (right) Scatter plot of average MCMC posterior variance vs. analytical gradient for samplings performed using these distances.

credible intervals is in the selection of candidate steps in the chains and measurement noise in the initial data. Seven laser offset distances are chosen, based on the gradient of the analytical solution as a function of this distance. These include the maximizing location and points nearer and further from the crack, including two locations having the same gradient magnitude. Five MCMC inverse solutions are performed for each of these locations, with the same five sets of noise in the initial data. The variances of the resulting posterior distributions are shown in Figure 2.7.

We find monotonic behavior in the variance of the estimation, decreasing with increasing gradient at the analytical solution evaluated at each laser offset. Furthermore, trials for offsets yielding the same gradient result in variances that are nearly identical, despite coming from laser positions both nearer and further than the maximizing position. These results support the optimization hypothesis we use, and suggest a deep connection governing this observed relationship

Parameter	Optimal Trend	Value Used
Laser offset	Depends on other parameters	1.56 mm
Pulse duration	Increasing (asymptotic)	0.5 s
Laser power	Increasing	10 W
Beam width	Decreasing (asymptotic)	1 mm
Material conductivity	Decreasing (asymptotic)	138 W/m K
Material density	Decreasing (asymptotic)	2680 kg/m^3
Material specific heat	Decreasing (asymptotic)	880 J/kg K

Table 2.1: Optimal trends for parameters in the crack characterization problem.

between estimate confidence and the gradient of the associated analytical solution. We proceed with an analysis optimizing the crack characterization inspection design, referring to *optimal parameters* as those which analytically produce maximal gradient.

2.4.2 Optimal Trends for Parameters

There are several modeling parameters in the mathematical formulation of the proposed laser pulse inspection modality. Some of these parameters can be controlled in an inspection, while others are properties of the material itself, or may be inherently fixed due to material selection. We report the optimal trends for all of the parameters in Table 2.1. These are found by numerically maximizing the gradient magnitude of Equation (2.2) with respect to each parameter individually. Some parameters increase the gradient asymptotically, giving small marginal increases as the parameter is increased. These are labeled "asymptotic" in Table 2.1.

The particular values which we use in the simulation of this inspection are also tabulated. These are chosen with consideration of the optimal trends, keep-

ing with realistic limitations. For example, the use of a more powerful laser would provide a clearer image, but quickly becomes cost-prohibitive and dangerous to use. Furthermore, parameters which asymptotically increase the thermal gradient are balanced between these dimishing gains and practical limitations. The behavior of the magnitude of the gradient function, as it depends on the pulse duration, is shown in Figure 2.8. A pulse duration of 0.5 seconds achieves 97% of asymptotic limit. The duration must be doubled to give another 1% increase towards the limit, so 0.5 seconds is chosen as the pulse duration for our idealized inspection design. The beam width is fixed at 1 mm for the reasons stated in Section 2.3.5, though we note that a tighter beam would offer some minor improvements in the inspection effectiveness. The angle made between the laser spot and the major axis of the crack is not included in the mathematical formulation. However, simulations have confirmed the intuition that the crack is most-easily detected when it is perpendicular to the flow of heat, a condition that is achievable using data from the first pass, as described in Section 4.2. Lastly, all of the modeling parameters specific to a given material are set according to values found in the literature for Al 5052.

The distance between the laser spot and the center of the crack has the most interesting behavior. The optimal offset of the laser was stated by Li et al. to be one laser radius [40]. We found a nonlinear relationship for optimal location, based on all of the other parameters. The experiments and numerical test in Reference [40] are done with few data points, linear interpolation, and their selection is done based on the best single result. With the method used here, any particular specimen, material, and/or laser can be treated. Using the parameter set specified above for our proposed inspection design, the analytical gradient is 14% higher than if a laser offset of 1 mm was used (i.e. if the criterion of Li et

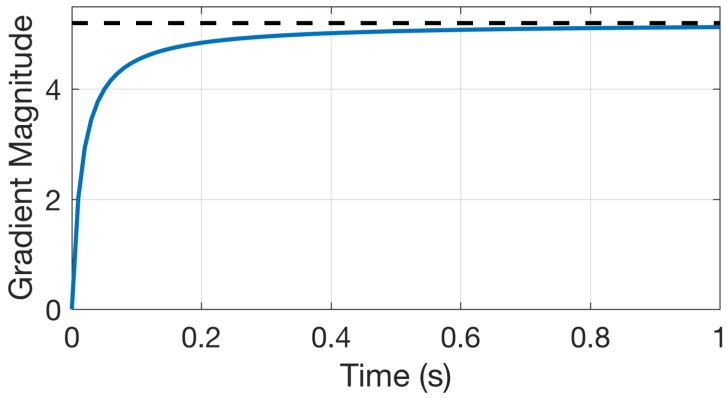

Figure 2.8: Maximal gradient magnitude as a function of pulse duration for an optimally-located laser spot as an example of asymptotic behavior. The dashed line denotes the limiting value.

al. had been applied).

2.4.3 Optimal Laser Position for a Bounded Domain

The analytical solution of Equation (2.2) over an infinite planar domain can be used to furnish a solution to the laser-heating problem over an uncracked bounded domain using the method of images described in Section 2.3.1. By considering a tessalation of Equation (2.2) centered at points symmetric over the domain boundary, there can be no net heat transfer along the line of symmetry, so as to satisfy the insulation condition specified in our problem description (Equation (4.1)). The effect of an insulated bounded domain is that heat accumulates on the boundary, causing an asymmetric thermal response, meaning: results display a dependency for the optimal laser offset based on the direction to the crack, as well as a bias to some directions (the gradient also loses its radial symmetry). Thus for a given crack location, we seek the optimal laser position within the domain, rather than simply the distance between the crack and the

laser peak. We explore this new problem defined on our 100 mm $\times$ 100 mm domain, independently considering 361 flaw locations uniformly spaced over the panel. For each, the laser peak position that maximizes the gradient magnitude of the approximated solution at that flaw location is found. The findings are summarized in Figure 2.9.

A pattern emerges from these results near the boundary, but further away from it, the insulating effect is lost, and all radial positions around the flaw are equivalent in term of optimal gradient (represented by a circle having the optimal laser offset as its radius). It seems for this scenario, the existence of a boundary in the domain has no effect on optimization analysis for cracks more than 20 mm away. This observation should extend to any sort of inhomogeneity within the test domain. Hence, other flaws, rivets, or actual panel boundaries, which are not perfectly insulated in practice, do not affect the validity of our optimization results, unless the flaw under study is particularly near them. In this case, optimal inspection parameters can still be determined numerically, provided that the forward model describes the effects of these local features.

2.4.4 MCMC-Based Inversion Results Using Optimal Experimental Conditions

Two sets of Markov chain Monte Carlo samplings are performed for a specific crack characterization problem. An elliptical crack with a semi-major axis of 0.25 mm and semi-minor axis of 0.03 mm is set in the center of a 100 mm $\times$ 100 mm aluminum 5052 panel. This is smaller than any crack that has previously been studied in the literature on thermographic imaging. Lasers with spot size

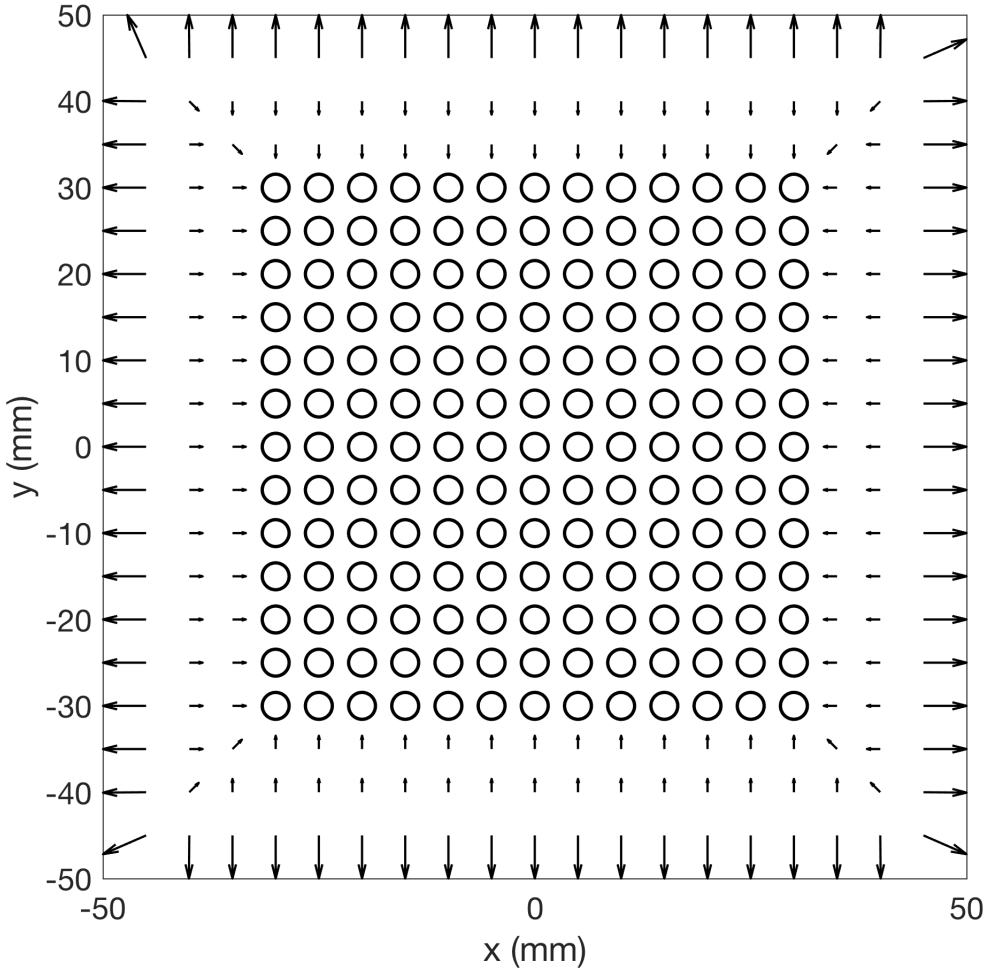

Figure 2.9: Vector diagram showing optimal laser location given a crack location over the 100 mm $\times$ 100 mm domain. The arrow bases are independently considered crack locations, with each arrow pointing at the optimal location for the laser peak to characterize a crack at its base. A circle denotes that a crack in its center is optimally characterized using a laser peak anywhere along its circumference: that is, asymmetry in the thermal gradient is negligible.

of 1 mm and heating times of 0.5 s are used in all samplings. Furthermore, the same thermal camera with pixel size, NETD, and thermal resolution stated above is considered. In the first set of 20 samplings (set 1), a laser power of 100 W is used, with 10 mm offset between the laser peak and the crack. This is done for comparison with reference to [17], as a benchmark representing previous related work from the literature. In the second set of simulations (set 2), the optimal distance between the laser and the crack, 1.57 mm, is used, and the laser power is set to only 10 W. This laser power is chosen to be similar to what was used in prior experiments with larger flaws as a reasonable-cost instrument [39, 40, 60, 32]. Additionally, the analytical gradient of the two sets of laser parameters are found to be close, 4.9 and 5.0, respectively, so they are expected to give similar confidence in their posterior estimation.

All posterior estimates here are gathered from MCMC sampling having 5,000 discarded burn-in steps and 20,000 saved samples. A separate study was done with a total of 300,000 samples, to affirm that these MCMC samplings are sufficiently long to believe that they are sampling from the actual stationary posterior probability densities of the various crack parameters. The methodology and results of this convergence study are detailed in A.3. The two simulation contexts have the same remaining underlying modeling parameters, each contaminated with i.i.d. Gaussian additive noise on the "true" data. These data are gathered from a separate finite element PDE solutions. Furthermore, each of the simulations are initialized at random values of the crack parameters of interest, taken from the support of the prior distribution.

We summarize the results of these MCMC samplings in three ways. First,

Crack Parameter	True Value	Set 1 Mean	Set 1 St. Dev	Set 2 Mean	Set 2 St. Dev
x location (mm)	0.0	0.05	0.12	-0.01	0.14
y location (mm)	0.0	0.02	0.24	0.02	0.20
Semi-major axis (mm)	0.25	0.25	0.03	0.24	0.03

Table 2.2: Aggregated mean and variance from two sets of 20 MCMC samplings.

the posterior statistics of the two sets of chains are directly compared. The mean and variance of each chain of 20,000 samples are aggregated, averaged over each sampling in the set, in Table 4.3. As we expect from the gradient of Equation 2.2 for each set, the two give nearly identical results. The standard deviations of each parameter estimate give insight into their relative importance. Since all three parameters are inferred from the same data, we see that the crack length has the strongest effect on the disruption of heat flow. After that, the location of the crack closer or further to the laser (x direction) is more important than its location in the perpendicular direction (y direction).

The form of the posterior distributions for the estimated parameters are also interesting. A strength of the MCMC method is that the solutions it provides for inverse problems have more information than just the statistics in Table 4.3. Since the chains sample from a probability distribution, the relative frequency of each value represents the probability that the noisy data were produced from that underlying model parameter value. Because of this, we are able to plot the histograms representing the estimated marginal distributions of each crack parameter in Figure 2.10. These are representative of the 40 total chains.

Finally, we present 100,000 total samples aggregated from ten independent MCMC samplings having the same additive noise and different random starting

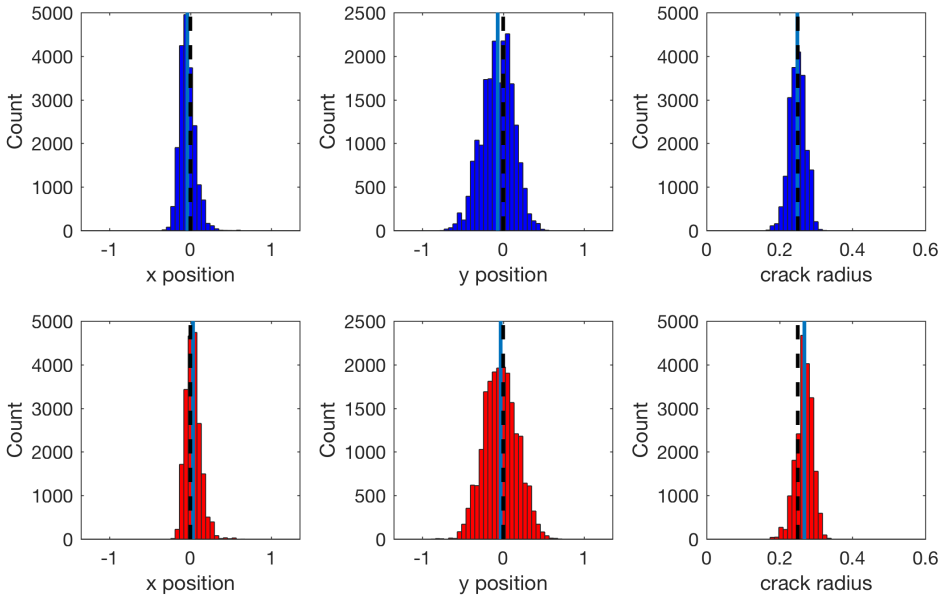

Figure 2.10: (upper) Marginal posterior histograms for one of the twenty MCMC samplings in set 1, and (lower) set 2. Solid vertical lines denote the means of the samples, while dashed lines denote the true values used to produce surrogate inspection data.

values (gathered for the convergence study in A.3). These samples are plotted in Figure 2.11 to show the relationship between the three parameters. Notable correlation is only seen between the crack length and the laser offset distance perpendicular to the major axis of the crack. This is because a small crack close to the laser spot will cause a disruption in the flow of heat similar to a larger crack further away.

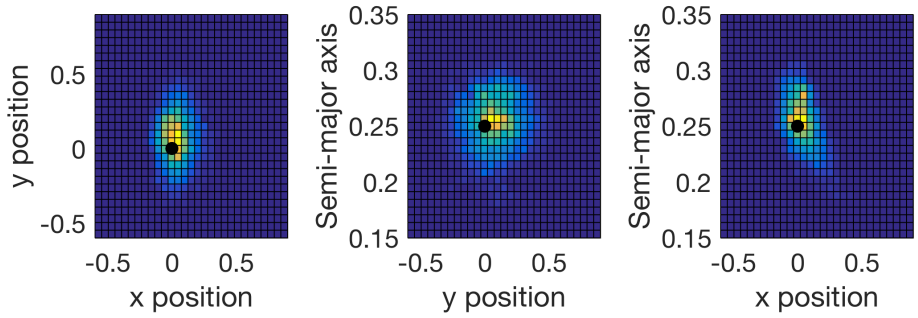

Figure 2.11: Joint posterior samples gathered from MCMC samplings used in the convergence study.

2.5 Conclusions

In this paper we set out a method for designing optimal non-contact thermographic inspection approaches for characterizing small through cracks in thin metal panels. The method relies on a two-pass active thermography inspection modality employing a laser and a thermal camera. After potential flaws have been located with a preliminary pass of the thermographic equipment, a brief second inspection is done for each crack. The specific setup of this second inspection requires choices for several design parameters. We have developed and presented a mathematical framework for describing the optimal choice of these parameters, so that each crack is as clearly revealed as possible within the resulting thermal data. This framework is validated using finite element simulations and tested in comparison with previous experiments in which the values of these parameters were chosen in other ways. It is shown that a 10 W laser directed at the location that is found to be optimal in our method provides data that are equivalent to earlier experiments in which a 100 W laser was required.

Although the specific problem of small through cracks in thin aluminum

panels is considered here, the framework for inspection design optimization that we describe is applicable for any metallic material. Furthermore, the particular MCMC solution to the inverse problem can be substituted by any other inverse solution method based on a user's time and accuracy constraints. The design parameters that our method produces are chosen to give data containing the most information about the underlying structure of the sample under evaluation.

CHAPTER 3

HETEROGENEOUS COMPUTING METHODS FOR SIMULATING HEAT CONDUCTION IN HETEROGENEOUS MATERIALS

3.1 Introduction

The finite element method (FEM) provides useful approximations to weak solutions of partial differential equations (PDEs) over a much broader family of domains than those for which analytical solutions are known [44]. The method involves transforming a continuum solution into the solution of a sparse system of linear equations, which are well understood and can be quickly solved or approximated. This transformation encodes the domain geometry as well as properties of the domain that can vary over space and time, such as material properties for heat conduction problems. Fast solvers for this type of problem are critical for providing simulations over domains with increasingly fine discretization. This is particularly salient in the case that many successive simulations are desired with changes to the parameterization of material coefficients, for example: as in the solution of a coefficient inverse problem [18].

The present study is motivated by such inverse problems. Furthermore, increasingly efficient methods for the simulation of heat transfer through heterogeneous material are useful for the design of structural and electrical systems [33]. In this area of research, analytical solutions have been found in some restricted cases [71], and only some characteristics of the transient behavior of more general problems are known [31, 4, 66]. The purely mathematical analysis relies on isotropy or bulk approximation over the domain. The presence of more than one set of heat conduction parameters over a heterogenous mate-

rial makes the problem too complex to solve, or results in solutions that are too complicated to be useful [31].

3.1.1 Scope and Organization

This paper is divided into six sections. In Section 4.2, some history of matrix-free methods for FEM is discussed; leading to recent heterogeneous computing approaches. In Section 4.3, the necessary mathematical preliminaries for our methods are summarized. The assembly operator is introduced, along with the flexibility that it offers for matrix-free methods. This flexibility is explored in Section 3.4 with three interpretations of the assembly operator and the resulting implementations for the simulation of heat transfer through heterogeneous media. In Section 4.5, numerical experiments and their results are described, to compare the speed of the implementations over a range of problem sizes. A sparse serial method is also included in the comparison. A coefficient inverse problem arising from a real-world need for nondestructive corrosion detection is introduced and analyzed in Section 4.4.5. Finally, Section 4.6 summarizes the goals and future impact of this research.

3.2 Background and Motivation

The methods that are developed in this work are derived from an element-by-element (EbE) decomposition of the finite element method. This viewpoint was introduced by Hughes et al. in 1983, for heat conduction calculations that were large for the time [29]. The motivation at that time was in avoiding exceedance

of available computer memory for the storage of the large FEM system matrix, rather than considerations of speed. The authors showed stability for the algorithm, and followed up to show the same properties of the EbE method for structural and solid mechanics problems [28]. Carey et al. exposed the potential for the EbE method to be parallelized, with a demonstration of a 2D convection-diffusion simulation in 1988 [11]. Following this, research interest in the EbE framework had little room for advancement until advances in computer hardware were made.

Heterogeneous computing is the practice of using dissimilar coprocessors in the solution of a numerical problem; typically a computer's central processing unit (CPU) and one or more graphics processing units (GPUs). This provides the user with access to fast serial processing power, as well as large-scale parallelism for certain computations that are properly amenable. Kiss et al. first utilized GPUs for the EbE method, using NVIDIA's proprietary CUDA platform [34]. The authors explored considerations that are particular to GPU computing, such as the preference for repeated computation rather than loading data from memory, and segmenting the domain with graph-coloring methods to avoid conflicts in parallelism. There have since been other efforts to parallelize PDE solvers for GPUs that are leveraged on symmetries of a particular problem [51].

Modifications to the classical EbE decomposition have recently been explored by Martínez-Frutos et al. [49]. The authors took a finer-grained approach to consider each degree of freedom (DoF) rather than a complete element, to formulate the DoF-by-DoF (DbD) method. It was shown in the context of elasticity problems that synchronization overhead from graph coloring is more costly than the unfavorable memory access patterns which it prevents, es-

pecially for 3D problems. Martínez-Frutos and Herrero-Pérez further explored the DbD method for elasticity problems on a fixed grid mesh so that only one local finite element matrix is required [47]. By varying material coefficients at the element level, between a constant "inside" and zero "outside", different domain geometries were enforced on the same mesh. More recent work by the same authors applied these strategies to problems of robust topology optimization [48]. By leveraging the ability to simulate many different domain geometries, an optimal structural design for a given set of loading conditions can be found. Heterogeneous computing methods for topology optimization have also been studied for problems of designing domains with desired thermal properties. These studies have shown good results for problems involving steady state heat transfer in 2D [70] and 3D [48], using FEM solutions to the elliptic time-independent heat equation. Lastly, Martínez-Frutos and Herrero-Pérez have shown that multiple GPUs can be used effectively in the solution of topology optimization problems through task-level parallelism—the simultaneous evaluation of independent models that arise within a collocation strategy. All of the heterogeneous computing research described above was done with the CUDA platform.

The present paper describes the theory and implementation of three approaches to a matrix-free preconditioned conjugate gradient algorithm for simulating transient heat conduction through a heterogeneous medium. Two are guided from previous studies, while our third combines benefits from both. The implementations differ in the interpretations of the DbD decomposition, as well as varying the use of a fixed grid or a general mesh, and in considering specialized hardware capabilities of GPUs. The most advanced implementation uses coalesced transactions with global memory for improvements in hardware ef-

ficiency, and is modified through a domain decomposition to run across dual GPUs. The domain decomposition is done in a way that minimizes communication between the two devices, so that the additional computational power can be effectively deployed. All of the implementation are made within the OpenCL computing framework, which is non-proprietary and free to use on any platform [64]. Scripting is done with the PyOpenCL package, providing readability and convenience with virtually no sacrifice in performance of the OpenCL API [35]. The performant code is available for public use, distribution, and modification [42].

3.3 Problem description

We present the mathematical and computational context for assembly-free finite element methods with a focus on the parabolic time-dependent heat equation PDE.

3.3.1 FE Formulation I (PDE)

We wish to solve the heat equation in three dimensions with spatially dependent material coefficients. In strong form, the boundary value problem is

$$\begin{cases} \rho C \frac{\partial T(\vec{x},t)}{\partial t} = \nabla \cdot (k \nabla T(\vec{x},t)) & \text{in domain } \Omega \times (0, t_f), \\ k \frac{\partial T(\vec{x},t)}{\partial \vec{n}} = f(\vec{x}) & \text{on sides,} \\ T(\vec{x},0) = T_{\text{ambient}}, \end{cases}$$

where the relevant thermal properties are the material density, ρ, specific heat, C, and thermal conductivity, k, all of which are assumed to be constant with

respect to temperature. Discretizing in time with a θ-scheme [44], the spatio-temporal temperature profile $T(\vec{x}, t)$ is reduced to a finite set of temperatures at regularly spaced time increments, $\{T^{(i)}(\vec{x})\}_{i\in\mathcal{I}}$, $\mathcal{I} = \{0, 1, \ldots, t_f/\Delta t\}$. The problem is then converted to weak form by multiplying the strong form with a test function $\phi(\vec{x})$ and integrating by parts to give the operators

$$a(T^{(i)}(\vec{x}), \phi(\vec{x})) = \int_\Omega \left(\rho C T^{(i)}\phi + \theta\Delta t k \nabla T^{(i)} \cdot \nabla\phi\right) d\vec{x}$$
$$L(\phi(\vec{x})) = \int_\Omega \left(\rho C T^{(i-1)}\phi - (1-\theta)\Delta t k \nabla T^{(i-1)} \cdot \nabla\phi\right) d\vec{x} + \int_{\partial\Omega} \Delta t f \phi ds.$$

We presume $T^{(i-1)}$ to be known, and find $T^{(i)}$ so that $a(T^{(i)}, \phi) = L(\phi)$ for all ϕ in some family of functions. If this family is composed of a finite set of basis functions $\{\phi_m\}_{m\in\{1,\ldots,N\}}$, the finite element method can be used to solve for $T^{(i)}$ as a linear combination of them: $T^{(i)} = \sum_m U_m^{(i)}\phi_m$, where $\vec{U}$ is a vector of coefficients. The discrete boundary integral of f is computed to give the vector $\vec{F}$. The finite element method involves the resulting matrices of pairwise integrals of basis functions

$$\mathbf{M} = \left[\int_\Omega \rho C \phi_m \hat{\phi}_n d\vec{x}\right]_{m,n\in\{1,\ldots,N\}} \text{ and } \mathbf{K} = \left[\int_\Omega k\nabla\phi_m \cdot \nabla\hat{\phi}_n d\vec{x}\right]_{m,n\in\{1,\ldots,N\}}.$$

Computation of these matrices is the process of *assembly*. With $\mathbf{M}$ and $\mathbf{K}$ available, solving the weak form of the heat equation for all $\phi \in \{\phi_m\}_{m\in\{1,\ldots,N\}}$ is equivalent to solving the matrix equation at each time step

$$[\mathbf{M} + \theta\Delta t\mathbf{K}]\,\vec{U}^{(i)} = [\mathbf{M} - (1-\theta)\Delta t\mathbf{K}]\,\vec{U}^{(i-1)} + \Delta t\vec{F}$$

for $\vec{U}^{(i)}$. This differs from the process of solving for a steady state heat flow in two ways. First, the FEM system matrix has a more complicated structure, rather than only involving the stiffness matrix $\mathbf{K}$. Second, the system must be solved at every time step to produce a transient solution. The time discretization process requires its own considerations for numerical accuracy and stability [44]. In this work we set $\theta = 0.5$, corresponding to a Crank-Nicolson method.

3.3.2 FE Formulation II (Assembly-Free Methods)

The matrices $\mathbf{M}$ and $\mathbf{K}$ are typically constructed by summing local contributions from each element in the assembly process. A local assembly matrix for element e, with D degrees of freedom, contains the pairwise inner products of all basis functions with support in element e,

$$\mathbf{M}_e = \left[\int_{\Omega_e} \phi_m \hat{\phi}_n d\vec{x} \right]_{m,n \in \{1,\ldots,D\}}.$$

Material property coefficients are taken to be constant over each element, so that the elemental assembly matrices depend only on the geometry of the domain. If all of the finite elements are the same size and shape, a single elemental assembly matrix can be reused and the mesh is said to have a *fixed grid* (FG). The *assembly operator,* A, over the index set of elements $\mathcal{E}$ denotes the process of constructing a full system matrix from its local contributions. For example,

$$\mathop{\mathsf{A}}_{e \in \mathcal{E}} (\rho C)_e \mathbf{M}_e = \mathbf{M}.$$

The assembly operator can also be applied to contributions within a single vector over each element to give the full vector. It is only a notational convenience to describe the mapping from local degrees of freedom to sums over global degrees of freedom. As such, the following are valid notation for the general expression $\mathbf{M}\vec{x} = \vec{y}$

$$\mathop{\mathsf{A}}_{e \in \mathcal{E}} (\rho C)_e \mathbf{M}_e \vec{x}_e = \mathop{\mathsf{A}}_{n \in \mathcal{N}} \left(\sum_{e \in \mathcal{E}^{(n)}} (\rho C)_e \mathbf{M}_e^n \vec{x}_e \right) = \vec{y}, \tag{3.1}$$

meaning that assembly is computed in terms of the degrees of freedom (over index set $\mathcal{N}$) in the "outer loop" with each of their elemental contributions computed separately. The first method is an EbE approach, similar to the standard method of assembling $\mathbf{M}$. The second is a DbD approach, in which the necessary vector dot products are viewed with finer granularity [49]. The freedom of

interpretation of the assembly operator gives rise to the different strategies for parallel matrix-vector multiplication that are described in Section 3.4.

To simplify notation, let $\mathbf{A} = [\mathbf{M} + \theta\Delta t\mathbf{K}]$ and $\mathbf{L} = [\mathbf{M} - (1-\theta)\Delta t\mathbf{K}]$. Set $\vec{b} = \mathbf{L}\vec{U}^{i-1} + \vec{F}$ and $\mathbf{A}_e = (\rho C)_e\mathbf{M}_e + \theta\Delta t k_e\mathbf{K}_e$ for $e \in \mathcal{E}$. Then the problem of finding $\vec{U}^i$ at each time step is reduced to solving

$$\mathbf{A}\vec{U}^i = \mathop{\mathsf{A}}_{e\in\mathcal{E}} \mathbf{A}_e\vec{U}_e^i = \mathop{\mathsf{A}}_{n\in\mathcal{N}} \left(\sum_{e\in\mathcal{E}^{(n)}} \mathbf{A}_e^n\vec{U}_e^i \right) = \vec{b}$$

We note once again that the explicit computation and storage of $\mathbf{A}$ and $\mathbf{L}$ is not necessary if $\{\mathbf{M}_e\}_{e\in\mathcal{E}}$ and $\{\mathbf{K}_e\}_{e\in\mathcal{E}}$ are available.

Assembly-free methods are especially useful if the spatially dependent material properties are not known in advance, or if many simulations are to be done over the same domain with varying coefficients. The generation of the mesh geometry and the computation of elemental assembly matrices can be done in advance and stored. Then all of the remaining computations required for a matrix-vector multiplication are parallelizable. Figure 4.1 illustrates a 3D FG mesh with tetrahedral elements and three sets of material properties parameterized by shading. Each of these spatially varying functions for the material properties have a simple functional form, and can be efficiently employed with elementary assembly matrices that are computed and stored beforehand, to treat changing analysis contexts under material property variation.

3.3.3 Preconditioned Conjugate Gradient

The matrix $\mathbf{A} = [\mathbf{M} + \theta\Delta t\mathbf{K}]$ is large, sparse, symmetric, and positive definite. The system above is thus solvable with the preconditioned conjugate gradient

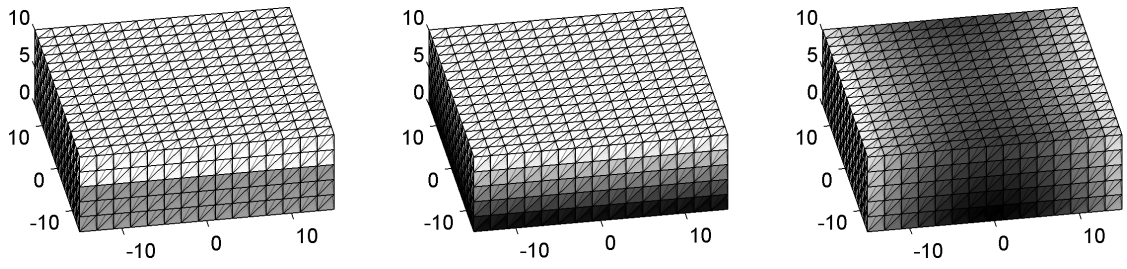

Figure 3.1: 30×30×10 mm domain with tetrahedral meshing and three functions describing changes in material properties. A boundary between two materials can be abrupt (left) or smoothed over many elements (middle). Heat conduction can also be simulated on a domain with more complex dependence between material properties and spatial location (right) with negligible computational burden, provided that the properties have a closed functional form (here, the shading is computed as $x^2 - 0.2y^2 + 10z$ at each vertex and the values averaged over each element).

(PCG) algorithm [63]. At each time step, let $\vec{x} = \vec{U}^{(i)}$ (to free superscripts and the index i), be set at some initial guess. There also must be provided $\mathbf{A}, \vec{b}$, a residual tolerance, and a preconditioner matrix $\mathbf{P}$ for which $\mathbf{P}^{-1}\vec{x}$ is easily computable and $\mathbf{P}^{-1}\mathbf{A}$ is relatively well conditioned, such as the Jacobi preconditioner. The algorithm involves the following linear algebra operations: matrix-vector multiplication (MVM), diagonal inverse matrix-vector multiplication (DIMVM), vector-vector multiplication (VVM), and adding scalar multiples of vectors (VAVSM). The dominating computation is the MVM in each iteration. All of the other computations are easily parallelizable. The parallelization and implementation of the MVM on GPUs for system matrices of heat conduction FEM problems is the focus of this work. The subsequent PCG algorithm is shown in Algorithm 1 with each step annotated by its type of linear algebra operation.

A Jacobi preconditioner is used for all of the methods described in this work

Algorithm 1: Preconditioned Conjugate Gradient

1: **function** PCG($A, b, x, i_{\max}$, tol, P)
2: $i \leftarrow 0$
3: $r \leftarrow b - Ax$ ▷ MVM
4: $d \leftarrow P^{-1}r$ ▷ DIMVM
5: $\delta_{\text{new}} \leftarrow r^T d$ ▷ VVM
6: **while** $(i < i_{\max})$ and $(\delta_{\text{new}} >$ tol$)$ **do**
7: $q \leftarrow Ad$ ▷ MVM
8: $\alpha \leftarrow \delta_{\text{new}}/(d^T q)$ ▷ VVM
9: $x \leftarrow x + \alpha d$ ▷ VAVSM
10: **if** i is divisible by 50 **then**
11: $r \leftarrow b - Ax$ ▷ MVM
12: **else**
13: $r \leftarrow r - \alpha q$ ▷ VAVSM
14: $s \leftarrow P^{-1}r$ ▷ DIMVM
15: $\delta_{\text{new}} \leftarrow r^T s$ ▷ VVM
16: $\beta \leftarrow \delta_{\text{new}}/\delta_{\text{old}}$
17: $d \leftarrow s + \beta d$ ▷ VAVSM
18: $i \leftarrow i + 1$

[63]. The construction of a Jacobi preconditioner is a straightforward process that lends itself to element-wise parallel computation. We note that the development of preconditioners that can be computed on a GPU is an active area of research [47, 23], but is not a focus of the present study. A comparison of this strategy with a serial implementation having a stronger preconditioner is made in Section 3.5.3.

3.3.4 OpenCL Heterogeneous Computing Framework

Computation hierarchy

With OpenCL, GPUs are programmed with *kernels*; small bits of C code that are sent in parallel to the individual cores [64]. At any given time, each of the many cores is acting as a *work item*, which is the most granular operating unit in the hierarchy. Work items are structured in *work groups*, with as few as one work item per work group. The user is responsible for defining the sizes and dimensions of the hierarchical structure, so that at the time of execution, each work item is provided with unique identifying information and the generic kernel code. The identifying information is:

- *global_id* ranging from 0 to the total number of work items in each dimension,
- *local_id* ranging from 0 to the number of work items in a work group, in each dimension,
- *group_id* ranging from 0 to the total number of work groups in each dimension.

The total number of work items, total number of work groups, and size of each work group is also available. The kernel explicitly tells each work item how to contextualize itself within the larger problem, to determine what data must be loaded from memory or computed privately. At the end of the kernel, results of the independent granular computations are written to memory.

Memory hierarchy

The use of memory on a GPU dictates programming strategies and the success or failure of an algorithm. At fully efficient throughput, a single core can execute several floating point operations per clock cycle [14, 16]. However, accessing data from the "slow" global memory location can take 400-600 clock cycles. This alone warrants special attention to the OpenCL memory hierarchy.

Data that is loaded onto the GPU, or stored as the output of work items, must be stored in *global* memory, which has space on the order of gigabytes. During execution, all work items have their own small amount of *private* memory, which is on-chip and not visible to any other work item. This is fast to access, and used for variables that can take different values across every work item. In between, there is *local* memory, which is also on-chip, and shared among a single work group. There are usually tens of kilobytes reserved for local memory for each work item. Access to local memory is roughly 100 times faster than accessing global memory, provided that work items within the work group aren't making conflicting calls ("bank conflict"). Local memory is allocated outside of the kernel, and cannot be freely initialized with specified data. Finally, kernels can consider certain data as *constant* memory. This data is physically still in global memory, but a kernel cannot write to it. When a kernel reads data from

constant memory, it is cached, so that subsequent reads are fast. Global memory is not cached.

There are two strategies for gracefully managing reads from global memory when it is necessary. First, the latency can effectively be hidden if there is enough non-dependent computation to keep a work item busy between the time when the data are called and the time they are used. This is preferable, though not always possible. Second, a kernel can take advantage of the way that hardware loads data from global memory to private memory. A single transaction with global memory yields 32 words (such as 8 byte double-precision floats) of data, whether it is all called for or not. If work items that are indexed sequentially by global ID, request data from global memory that is organized in the same sequential way, the calls are automatically bundled and processed as one transaction. This process is the simplest form of *coalesced* memory access. There have been advances in hardware, and in OpenCL standards, to provide more flexibility, such as allowing permutations of the 32 sequential words to 32 work items, that are blocked together but not necessarily in the same order.

Programming strategy

The OpenCL programming approach is as follows:

1. Investigate the hardware to find and define a *host* (CPU/hard drive etc., the conventional computing environment), its *devices(s)* (GPU with its onboard memory), and define a *context* the overall computing environment.
2. Define initial variables on the host
3. Load data onto a device. This includes reserving space for data that a

kernel will write later and anything that is meant to last from one kernel to another. The benefit of defining all of the memory space in advance is that the user may specify whether each buffer is read or write only (or both) for both the host and device, or even if it is known that the host will never try to read it. Then the space that is allocated will be optimal for however the data will be treated.

4. *Build* the compute kernels. This involves compiling the C code and specifying pointers to memory buffers where its arguments can be found. A single kernel program can be built multiple times with different arguments, as is the case with the vector-vector multiplications in steps 5, 8, and 15 in Algorithm 1. Each of these are built independently.

5. Define a *queue* in the context. The host can enqueue kernels, memory transfer operations, or wait fences. OpenCL turns these into individual tasks that are performed as cores on the GPU become available. Code is written as if the context has infinitely many cores to run in parallel, and then the queue manages the execution of code on available hardware. Flags can be used with enqueued commands to ensure that all tasks from one kernel are finished before any tasks from the next kernel start, in case memory is being written and then read in a dependent way.

6. Enqueue commands to copy memory from a device to the host. This can be the final result of computations, or in the case of this work, the PCG residual, so that the host can decide whether or not to begin another iteration of enqueueing kernel commands.

3.4 Implementation of Assembly-Free Methods

We outline three assembly-free algorithms for matrix-vector multiplications. The differences arise from the flexibility in interpreting the assembly operator demonstrated in Equation (3.1). For each method, the explicit assembly equation is provided, along with an outline of the memory and computational hierarchy for a parallel implementation on GPUs. Further details are discussed in B.1 so that broader concepts behind the implementations can be the present focus. We begin by giving context of the particular geometry of the problem.

3.4.1 Mesh Geometry

A 3D regular mesh with linear tetrahedral elements is generated based on the domain boundaries and the number of divisions in each dimension. The domain is then divided into rectangular prisms according to these divisions. The examples here are all cubes for simplicity. Each cube is then subdivided into six tetrahedra, as shown in Figure 4.1. A benefit of using tetrahedral elements is that their elemental assembly matrices have size 4×4. The OpenCL specification allows dot products of the 4 element floating point vector data type, *float4*, with a single instruction. This is the central operation for all of our fast MVM methods, regardless of the interpretation of the assembly operator.

The six tetrahedra within each cube are indexed in a sequential way—six in the first cube, then six in the next cube in the x direction, etc. until the y dimension is incremented, and then the next slice in the z dimension begins after that. An emphasized view of the six tetrahedra is given in Figure 3.2. Each

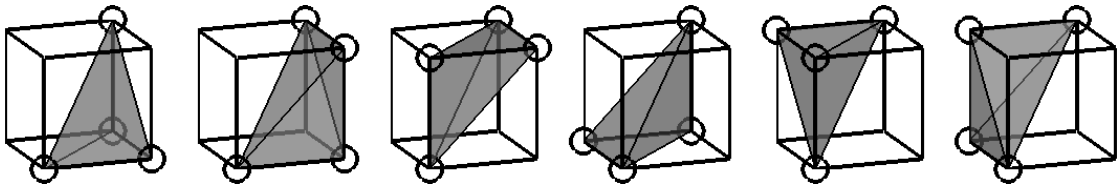

Figure 3.2: Emphasized view of the tetrahedral subdivisions of each small cube in the mesh. Each finite element is characterized by four vertices of the cube.

cube has consistent indexing as to which of its eight corners correspond to each element within; allowing for a single reference table to be stored in memory. The structure of this kind of mesh also permits local computation, such as determining spatially dependent material properties, without the need for a table of xyz-locations of each vertex. This information can be determined from the one-dimensional index of the vertex along with a small table of the domain boundaries and the number of divisions in each dimension. Substituting small computation from general domain information, in place of loading the same data from a lookup table, is important for parallelizing an algorithm for a GPU.

3.4.2 Previous Strategies

We begin by providing the details of two approaches that use ideas from previous literature.

Implementation 1: General DbD

The first implementation follows a general DbD strategy that does not assume that the FEM mesh lies on a fixed grid. Each global DoF of the vector $\vec{y} = \mathbf{A}\vec{x}$ in

a general matrix equation is computed as

$$\vec{y}_i = \left[\sum_{\substack{e \in \mathcal{E}^{(i)} \\ i = \mathcal{N}^{(e)}(j)}} \left[\mathbf{A}_e^j \vec{x}_e \right] \right] \quad (i \in \mathcal{N}), \tag{3.2}$$

where each pair of square brackets denotes a computation done by one work item in the implementation. Every elemental assembly matrix is stored in global memory, with a preprocessing step that determines $\{\mathbf{A}_e\}_{e \in \mathcal{E}}$ and $\{\mathbf{L}_e\}_{e \in \mathcal{E}}$ based on local material properties. A subset of this computation furnishes the Jacobi preconditioner matrix at the same time, by only storing diagonal entries. Work groups are responsible for each set of six elements in a cube, with 24 work items per work group, each corresponding to one element-DoF pair. Elemental assembly matrix coefficients are computed one time, and the scaled matrices are stored in float4 vectors in "element order" within global memory. In the computation, each of the 24 work items loads vertex data to local memory, including duplicated vertices, to alleviate conflicting simultaneous memory access from multiple work items. Then it reads its row from the elemental assembly matrix data, takes necessary entries from local vector data, performs a dot product, and stores the result as its contribution to the global vertex in "vertex order". In a second pass, one work group for each global vertex reads these 24 consecutive contribution and sums them into the single entry of the result vector. A similar two step approach is described by by Martínez-Frutos et al. as an alternative to atomic incrementation to a single memory location for each degree of freedom [49].

The necessary data and computational responsibilities of the first pass of this process are illustrated in Figure 3.3. The left panel shows the scope of each work group—six elemental assembly matrices and eight entries of the input vector $\vec{x}$, which are loaded into the work group's local memory. The right panel

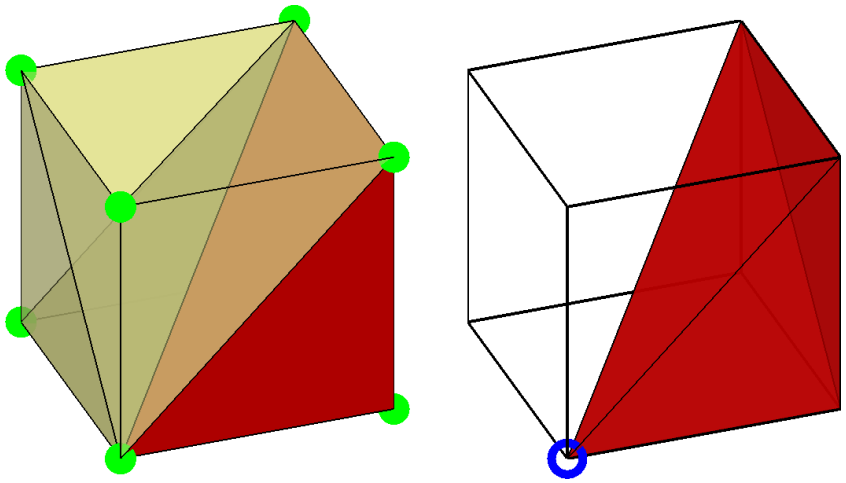

Figure 3.3: (left) Local data requirements for the first assembly-free MVM method. Elemental assembly matrices are required for six tetrahedral elements that comprise a cube in the mesh, as well as entries of the input vector corresponding to its eight corners. One element is emphasized and isolated (right), denoting the responsibilities of a single work item. There are 24 work items, within each work group for all such finite element-DoF pairs, required to assemble the output vector.

illustrates the responsibilities of each work item—one element-DoF pair, corresponding to a length-four vector dot product that it must compute and store. For clarity and consistency with the following sections, the necessary data for this single work item is emphasized in the left panel as well.

Splitting the assembly operator into two explicit sums requires a sort of data transpose at some point of the computation. This is because assembly matrix data is naturally stored in element order, while the output must be combined and stored in vertex order. The two step approach here ensures efficient data retrieved from global memory for both steps. The consequences of writing data to potentially distant locations at the end of the first step are hidden, since the data is enqueued to be written, and then the kernel can be restarted and the computation continues on while the writing takes place. Furthermore, we note

the second step can be slightly modified to perform an extra vector-vector addition of the form $\vec{y} = \mathbf{A}\vec{x} + \vec{b}$ in the same kernel, which is one of the operations of the PCG algorithm.

Implementation 2: Single Pass FG DbD

The second implementation follows the coarser parsing of the assembly operator, with

$$\vec{y}_i = \left[\sum_{\substack{e \in \mathcal{E}^{(i)} \\ i = \mathcal{N}^{(e)}(j)}} \mathbf{A}_e^j \vec{x}_e \right] \quad (i \in \mathcal{N}). \tag{3.3}$$

Once again, the square brackets denote the computations of a single work item, so that this method only requires a single pass. Single pass, fixed grid DbD strategies have been previously explored in References [47, 50]. The trade off here is that this requires more data to be loaded into local memory for each work group for the full determination of a degree of freedom. Consequently, more memory must be loaded overall. This is partially alleviated by setting the work groups to be as large as allowed by hardware limitations, so that most of the data are reused by adjacent degrees of freedom. Previous strategies to maximize on-chip memory have organized the input data into 2D square patches, which can be loaded from global memory in a coalescent way [61]. However, in order to have access to the data in neighboring elements, a halo of data around the patches must be loaded inefficiently. We choose to organize the necessary input data into 3×3 rectangular blocks, and as long as possible, so that all global memory access can be coalesced in the long direction, as shown in Figure 3.4. For our hardware, specified in Section 4.5, this corresponds to 64 work items and 27 KB of local memory usage. We also implement this approach to

be used on a fixed grid mesh so that only one elemental assembly matrix is required. This allows the elemental assembly matrix to be stored in constant memory, so that it does not need to be loaded anew by each work item. The Jacobi preconditioner is computed in the same way as was described in Section 3.4.2, although the local material properties are not precomputed. This is not necessarily detrimental, since computing local material scaling coefficients is a small computation which can hide the latency of loading data from the input vector.

A visual overview of this method is provided in Figure 3.4. Each work item is responsible for all contributions to one entry in the output vector, so it needs data from 24 local assembly matrices and 27 entries of the input vector. An internal loop scans through all 24 element-DoF contributions, computing elemental scaling coefficients, performing dot products, and cumulatively summing the result.

3.4.3 Implementation 3: FG DbD with Memory Coalescing

The third implementation combines advantages from the first two. It is similar in structure to the first, except with increased responsibility to each work item, larger work groups, and the restriction to a fixed grid. The explicit interpretation of the assembly operator for this method is the same as Equation (3.2), although the full elemental assembly matrix-elemental vector multiplication is carried out by a single work item here. The primary characteristic of this third method is that work groups are structured so that all data that is loaded from global memory in the first pass is done so with a coalesced access pattern, as de-

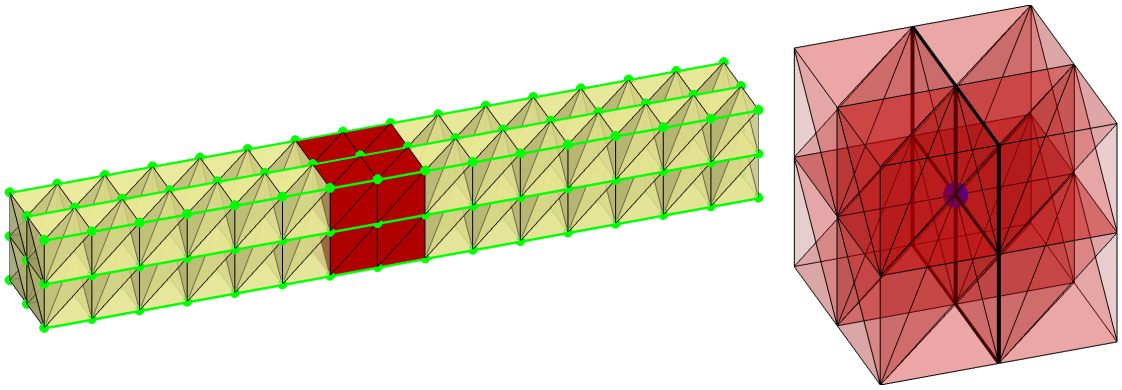

Figure 3.4: Necessary local data and work item responsibilities for the second assembly-free MVM method. (left) Data that is loaded from adjacent locations in memory for the input vector are connected by a green line to emphasize the potential for coalesced memory loading. Note that this figure is truncated for clarity and that the method actually loads 64 consecutive entries to local memory. (right) A single work item requires more data, but fully computes an entry of the output vector, denoted by a filled point. As before, the data required by a single representative work item is emphasized from local memory on the left.

scribed in Section 3.3.4. In this sense, it is similar to the second implementation, except that the work group size is determined by the amount of data that can be loaded from a single coalesced memory read rather than by maximum local memory capacity. Furthermore, only four sections of memory are required for the first pass, as shown in Figure 3.5. The element-DoF contributions to each entry of the output vector are collected and summed as much as possible before writing back to global memory. While a second pass is still required to add the contributions among work groups, this process is faster than in the first implementation since there are fewer terms to sum. Additional details behind the efficient use of coalesced memory access now follow.

Since a fixed grid is assumed, the only data that must be loaded from global

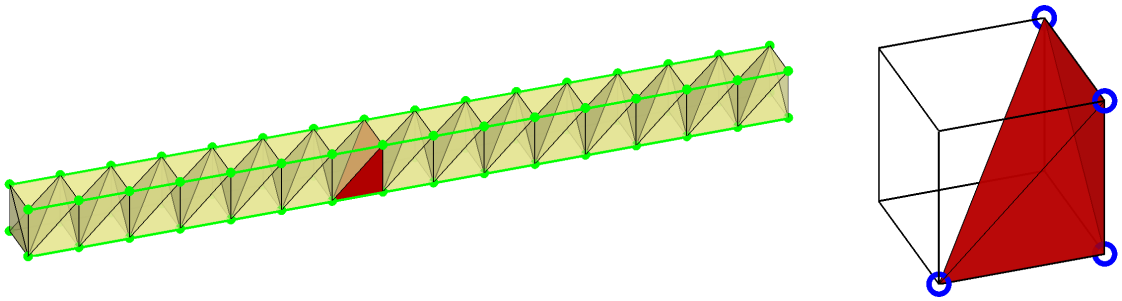

Figure 3.5: Necessary local data and work item responsibilities for the third assembly-free MVM method. (left) Four coalesced reads from global memory provide all of the necessary input vector data. Note that this figure is truncated for clarity and that the method actually reads 32 consecutive entries to local memory with each coalesced memory read. (right) The representative work item computes contributions for all degrees of freedom associated with its finite element.

memory is the input vector. To do this efficiently, it must be loaded in blocks of 32 consecutive entries by blocks of 32 consecutive work items. A single memory read of this form does not give enough data to perform assembly for any element. However, if four blocks of memory are read, corresponding to the four horizontal edges of a long rectangular prism, then every tetrahedral element within that prism can be integrated over. That is, the contributions from each of these elements to the degrees of freedom that are loaded can be computed.

Using the memory access length of 32 as the guiding limit, 31 cubes of six tetrahedral elements will be integrable. Therefore, work groups of 186 work items are invoked—one work item for each tetrahedral element. Since only 128 work items are necessary to read data from global memory, some work items are assigned a dual purpose and some remain idle during the loading process. Those first 128 receive both a global elemental index for their integration responsibility as well as a global vertex index for loading data from global memory

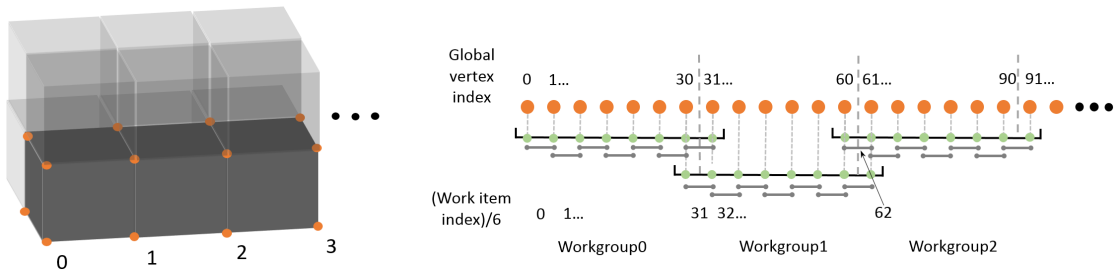

Figure 3.6: Memory access and computational partitioning pattern for the coalesced DbD MVM method. The orange nodes correspond to data in global memory, while the green nodes represent local memory in each work group. The short gray lines within each work group represent the tetrahedral elements which must have access to the vertex data at all four corners to compute their assembly contributions. Four such coalesced reads from global memory are performed to provide the tetrahedra with their necessary data.

into local memory. Figure 3.6 depicts the way in which global data is accessed by each work item for one of the four edges of the long rectangular prism. The other edges are treated in the same way, with offset information determined in-kernel based on how many divisions are made in the domain in each dimension. The contribution to a vertex can only be computed if all four vertices of an adjacent element are included within a work group. Therefore, only the 30 internal vertices from each block of 32 is contributed to in storage (except for the first work group, and possibly the last). This partitioning pattern is also demonstrated in Figure 3.6.

Another consequence of coalesced memory access is that some consecutive elements of the input vector do not actually share an element in the domain. This is demonstrated in Figure 3.7 for a small example mesh. We avoid this problem by padding the domain with non-physical elements in the $+x$ and $+y$ directions from the perspectives of loading and computation, and then the con-

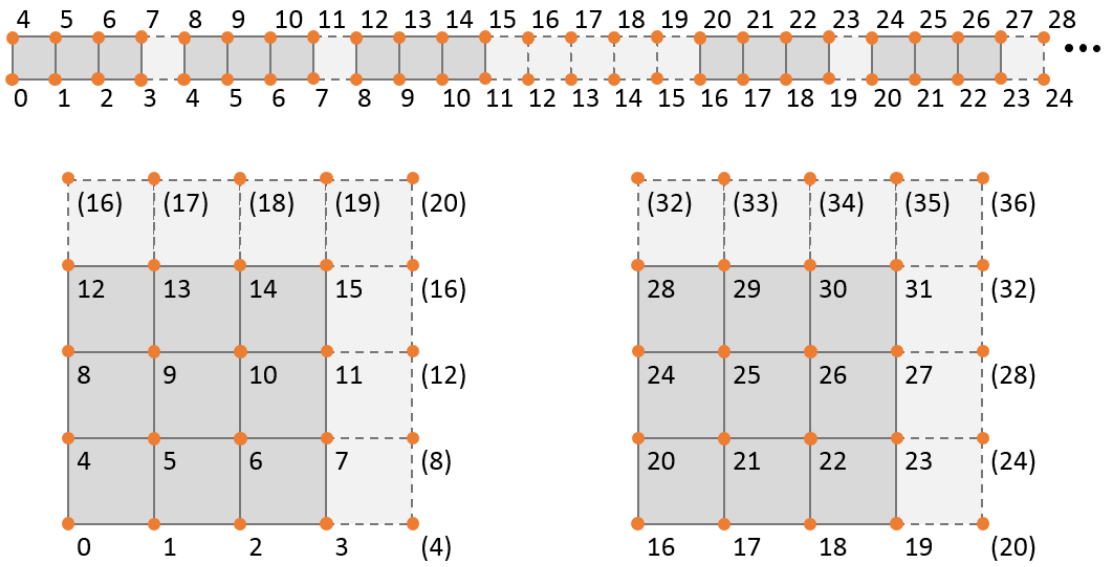

Figure 3.7: Element padding for a 3×3×n mesh as seen from a top-down perspective. All of the data is loaded, but only contributions from solid elements are stored. This allows coalesced access to global memory to be used without interruption, at the cost of the extra discarded computation.

tributions of these elements are ignored upon storage of the result. This produces some amount of wasted computation, but the relative volume of padding elements to the total mesh volume decreases as the granularity of the mesh increases. Furthermore, efficiency gains from coalesced memory access strongly outweigh the losses from this wasted computation.

3.4.4 FG DbD with Memory Coalescing on multiple GPUs

The fixed grid coalesced method described above is modified for use on dual GPU. The domain is split in the z direction according to the additive Schwartz method [10]. This keeps all vertices in each subdomain in adjacent blocks of memory. A fraction of the domain to be assigned to device 1, m, is specified,

such as 0.5. This fraction of z slices, rounded up, with one additional layer, is the number of vertices that device 1 receives for computing, m_1. The rest of the vertices, in addition to two overlapping layers are assigned to device 2, totaling m_2. This split is demonstrated in Figure 3.8. If input vectors are initialized from a full set of global data, then one matrix-vector multiplication can be performed on each device, and the result can be faithfully reconstructed. For more than one sequential matrix-vector multiplication, the shared boundary data must be updated. Only one layer of vertices needs to be transferred in each direction. In the PCG algorithm, the solution vector x is initialized at the beginning, and the intermediate vector d must be transferred at each iteration. This memory transfer presents a bottleneck in the method, so that gains in speed are expected only with large systems for which MVM takes much longer than a GPU-to-GPU memory copy.

The scalar results of dot products must also be communicated between devices, both the residual δ and step size α. To do this, each partial dot product is handled separately, omitting the extra boundary vertices. The partial results are stored in their own buffers, which are then transferred both directions. We thus require four memory buffers for each scalar quantity, one native buffers for the partial results on the device which computed it, and one target buffer on each device to store the copied value from the other. Many permutations of methods for the bidirectional communication of partial dot products were considered. This method was found to be the fastest reliable way for each device to receive the full results. No special kernel is needed to combine them. Rather, modifications to VAVSM are made to accept both buffers on a device and sum them as part of the existing process.

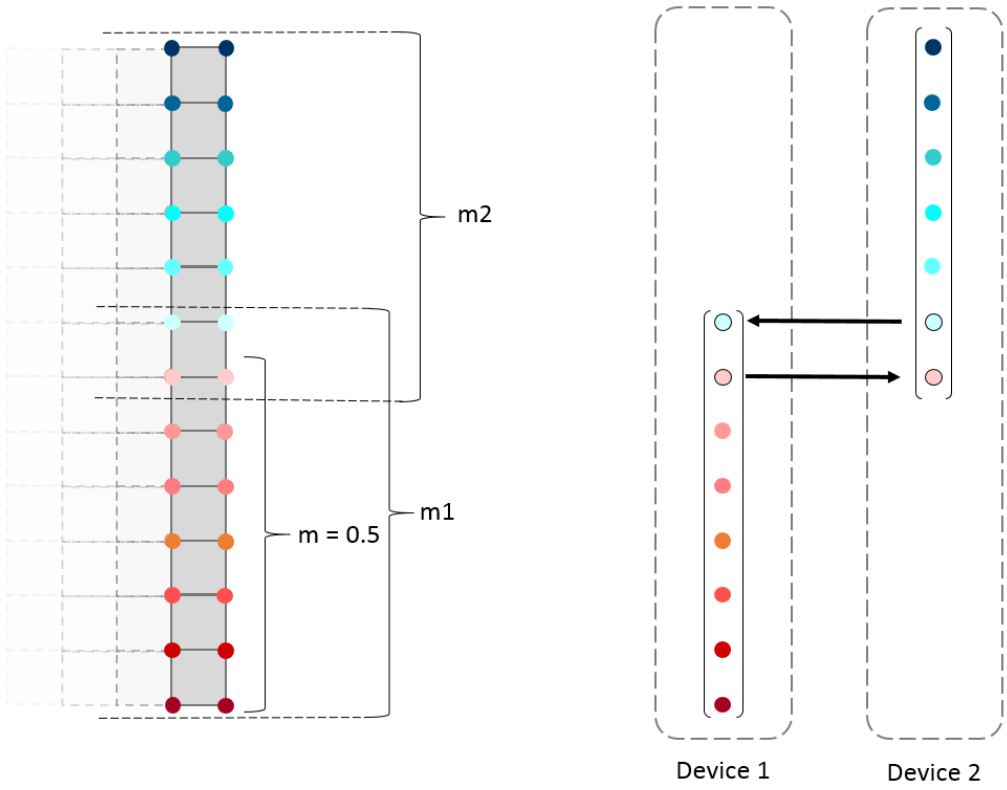

Figure 3.8: (left) splitting of vertices between two devices according to a user-specified fraction m. (right) Data that must be transfered between devices at each iteration.

The only remaining matter for modification is adjusting the memory objects that contain vertex-to-spatial-location information. One each is made for the two devices, with the second encoding the z shift for its first index. Then each devices runs PCG in parallel, and can contextualize the location of vertices within the total domain. The host initialized two sets of every kernel that has been described for the serial method, as well as launching one queue to transfer boundary and scalar information and properly wait. Lastly, we note that the implementation here is for dual GPUs, but in principle could be easily extended for increased distribution: with multiple GPUs.

3.5 Experiments and Discussion

The results shown here have been produced on AMD FirePro D700 GPU with 2048 streaming processors, 6GB of onboard memory, and up to 32KB of local memory per work group. The serial computations are done on a 2.7 GHz Intel Xeon E5 processor. Elemental assembly matrices are precomputed so that assembly of the sparse system matrix can be done efficiently after the specification of material parameters, analogously to the parallel GPU algorithms. Furthermore, Jacobi preconditioning is specified so that the comparisons below are as fair as possible.

3.5.1 Performance Comparison

We first report the performance of each implementation over a range of problem sizes by simulating uniform heating on the front face of a two-layer laminate. The domain for this problem is a rectangular prism $\Omega = [-15, 15] \times [-15, 15] \times [0, 10]$ with $f(\vec{x}) = 1$ on $x_3 = 0$ and zero everywhere else, and $T_{\text{ambient}} = 0$. The material parameters are specified as

$$\rho C = \begin{cases} 3.724\text{e}6\ \text{g/mm C s}^2 & x_3 \leq 5 \\ 1.65\text{e}6\ \text{g/mm C s}^2 & x_3 > 5 \end{cases}, \qquad k = \begin{cases} 4.9\text{e}8\ \text{mm}^2\text{C s}^2 & x_3 \leq 5 \\ 4\text{e}6\ \text{mm}^2\text{C s}^2 & x_3 > 5 \end{cases},$$

corresponding to mild carbon steel and its solid corrosion products, assumed to be iron (III) oxide, Fe_2O_3. We perform 50 PCG solutions with $\Delta t = 0.01$ seconds and a relative residual tolerance of 10^{-6}. We vary the density of the tetrahedral mesh over the rectangular prism and measure the walk clock time per PCG iteration for each method. This provides a performance metric that accounts

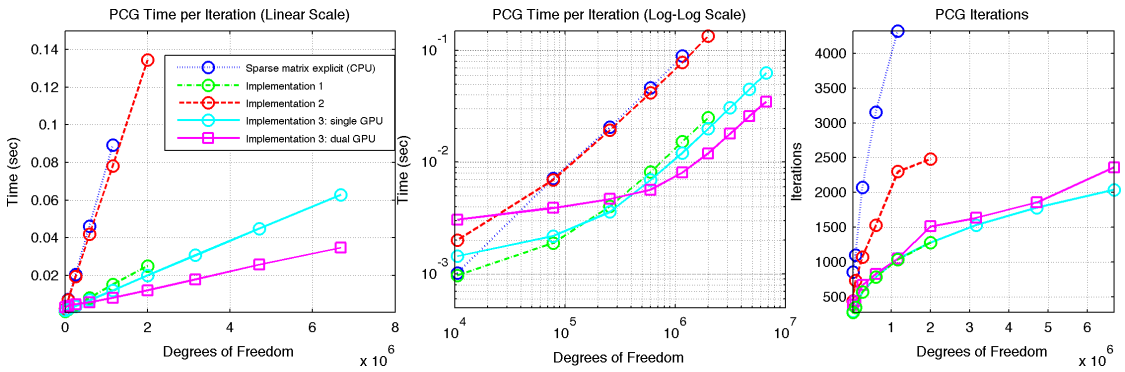

Figure 3.9: (left and center) Time per PCG iteration for four GPU implementations and a serial sparse implementation of the same algorithm. Results are computed for each method as far as hardware limitations would permit. (right) Total number of iterations required to solve a transient heat conduction boundary value problem with 50 time steps.

for computation time as well as the transfer of memory, and is consistent with the reporting of similar experiments [49, 47]. The results of these experiments are shown in Figure 3.9. Although there is sublinear progression with coarse meshes as memory transfer time dominates, all of the implementations exhibit linear increase in computation with the number of degrees of freedom. The rate of increase of wall clock time for each GPU implementation, taken from the linear asymptote, are compared with the sparse CPU method in Table 3.1.

Method	1	2	3: single GPU	3: dual GPU
Speedup Factor	6.5	1.2	8.4	16.8

Table 3.1: Ratio of the rate of increase of wall clock time with increasing degrees of freedom between the sparse matrix CPU method and the linear asymptote of each GPU method.

The experimentally observed linear scaling in computation with the number of degrees of freedom is expected based on the algorithms for computing MVM. Furthermore, it is seen that the efficiency of an implementation is di-

rectly related to the its care in the treatment of loading data from global memory. The second implementation has advantages over the first in that it does not require loading elementary matrix data or a second pass to sum contributions to a MVM. However, the cost is that it involves loading nine separate strips of data from the input vector to provide full information about all elements that are adjacent to a vertex. We see that this overhead outweighs the other merits of the implementation. The third implementation successfully adapts the memory management benefits of the previous two. A MVM requires only four strips of data from the input vector for the first pass, which are coalesced into four reads from global memory, and the contributions are partially summed so that the second pass will have less work. For the smallest problems considered here, the sparse matrix method is fastest, and the implementation on dual GPUs is the slowest. Further investigation into the dual GPUs implementation is done in the following section.

We also include, for comparison, the same profiling results when the GPU methods are set to compute with single-precision floating point arithmetic. As seen in Figure 3.10, all of the methods perform faster. We note that the point at which the dual GPU implementation becomes feasible is delayed to roughly double the problem size, as the balance between computation and memory transfer time is shifted.

3.5.2 Multiple GPUs

Performing the third PCG method on dual GPUs gives better performance for large problems, as expected. We see in Figure 3.9 that for problems with fewer

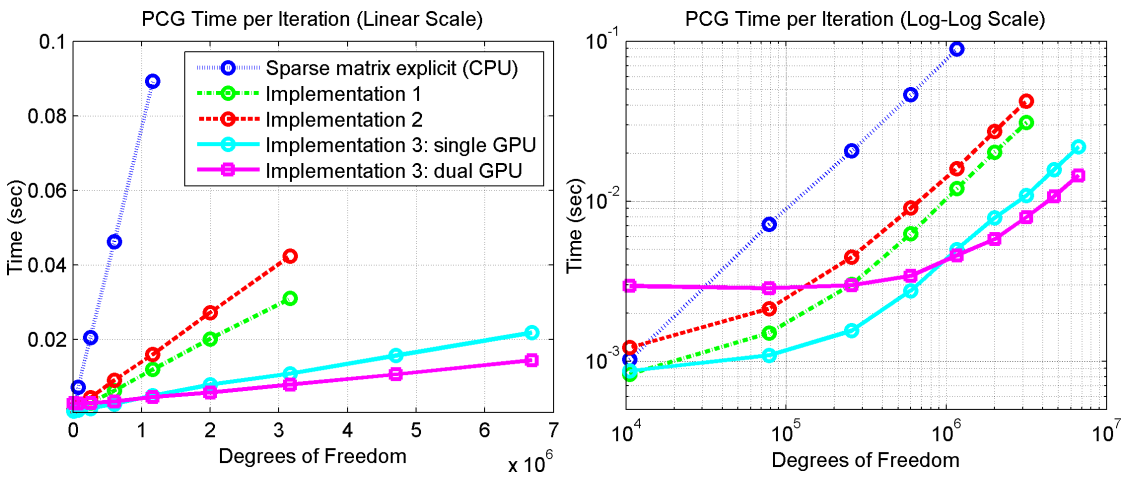

Figure 3.10: Time per PCG iteration for four single-precision GPU implementations and the (double-precision) serial sparse implementation of the same algorithm for comparison. Results are computed for each method as far as hardware limitations would permit.

than about 400,000 degrees of freedom, the increased memory transfer costs dominate total computation time. For larger problem, the burden of computing MVMs grows, so that increased parallelism is worth these costs. We use OpenCL profiling operations to confirm that the time spent actively computing MVMs is reduced by the same factor as the workload sharing (including the overlapping region), even though the directly observed total time per iteration exhibits smaller reduction. For problems larger than those considered here, it is expected that even more GPUs become practical, and we note that the extension is straightforward with the additive Schwartz method for domain decomposition.

3.5.3 Further CPU Comparison

We continue with a deeper comparison between our best performing heterogeneous computing method with a more sophisticated serial sparse matrix method. Since a standard CPU sparse matrix method can access the system matrix quickly, it is feasible to compute a better preconditioner matrix than the Jacobi preconditioner. This can take more time to produce and compute inverse matrix-vector multiplications with, but can vastly reduce the number of iterations required. We find that an incomplete Cholesky factorization with drop tolerance of 10^{-3} performs well for the simulations discussed in this paper [24]. Table 3.2 shows the benefits of using this preconditioner for a transient heat conduction boundary value problem over 50 time steps. Since it is no longer fair to compare the average time per PCG iteration, we report the total time that both methods take to provide a solution, starting from knowledge of the elemental assembly matrices and a parameterization of material properties, including the incomplete Cholesky factorization for the serial method. The effective time per iteration is determined from this adjusted total time, rather than exclusively from the time in the PCG outer loop.

Degrees of Freedom	10.5×10^3	78.1×10^3	257×10^3	600×10^3	1.16×10^6	2.00×10^6
Sparse matrix CPU: Total time (s)	0.9	1.9	7.7	21	51	97
Total Iterations	70	70	87	111	126	143
Time per Iteration (s)	13×10^{-3}	27×10^{-3}	88×10^{-3}	190×10^{-3}	400×10^{-3}	680×10^{-3}
FG DbD: Total time (s)	0.41	0.75	2.0	5.4	12.7	25.5
Total Iterations	287	344	567	780	1047	1278
Time per Iteration (s)	1.4×10^{-3}	2.2×10^{-3}	3.6×10^{-3}	6.9×10^{-3}	12×10^{-3}	20×10^{-3}

Table 3.2: Full performance comparison between the sparse matrix CPU method with incomplete Cholesky preconditioning and our best performing implementation.

We see that the sparse matrix CPU method with incomplete Cholesky preconditioning takes similar overall time for modestly large systems. However, the differences in wall clock time diverge as explicit storage of the system matrix becomes more demanding. The heterogeneous computing method developed in this research outperform even sophisticated serial algorithms.

3.5.4 3D Coefficient Inverse Problem

An important benefit of the currently proposed methods is that many simulations of heat conduction can be rapidly performed, in sequence, over a domain with varying thermal properties. This scenario arises in the solution of coefficient inverse problems, such as determining internal properties of a structure from noisy temperature measurements at its surface, in response to a known energy input. We explore such a case that has been earlier studied in two dimensions, motivated by a real-world corrosion detection problem [18]. By extending previous analysis methods from the literature to a full 3D model, we not only approach a more physically realistic use case, but are also able to relax certain symmetry restrictions on the boundary heat flux f. As a result, we found that an internal corrosion profile can be recovered with higher confidence, while using a heat source only one tenth as powerful as what was required in earlier work [18]. The details of the numerical experiment are now discussed.

Corrosion may form in a bridge structure in the crevice where a lower truss chord member meets a flat steel gusset plate connection element. It is observed that this corrosion within the gusset plate has a well defined geometric form, constant along the horizontal length of the truss chord, and varying as a

quadratic function in the vertical direction [18]. The severity of the corrosion is parameterized by the penetration depth of the apex of the parabola into the steel, θ, illustrated in Figure 3.11. We consider a case of corrosion that has penetrated 3.175 mm into a plate that is 12.7 mm thick: corresponding to a 25% section loss. The relevant thermal properties are the same as in Section 3.5.1: A thermal input is furnished in the form of a 10 W laser beam having a Gaussian profile with 2 mm beam width. The structure is heated for $T_F = 10$ seconds, after which the surface temperature is recorded with an thermal camera. Surrogate field data, $\mathcal{D}$, are created with a high fidelity forward model (1.16 million degrees of freedom and 0.01 second time steps), interpolating the FEM solution to a finer rectangular grid, averaging the temperatures over each pixel area, and contaminating the result with noise; all so as to approximate a plausible digital thermal camera measuring device. For consistency with previous work, the camera is assumed to follow a noise model consisting of contaminating each pixel with independent and identically distributed Gaussian noise with zero mean and standard deviation of 0.1 °C, then rounding the result to the nearest 0.1 °C.

The inverse problem is solved with Markov chain Monte Carlo (MCMC), a Bayesian inference method [25]. The solution comes in the form of dependent samples taken from a probability distribution over the corrosion penetration depth, $p(\theta|\mathcal{D})$, known as the *posterior* distribution. Since there is randomness in the surface temperature measurements, MCMC is able to automatically propagate a measure of uncertainty to the posterior distribution. The mean and standard deviation of a large number of posterior samples provide useful information about the underlying corrosion depth. The consequence is that every sample comes at the cost of performing a simulation with the candidate cor-

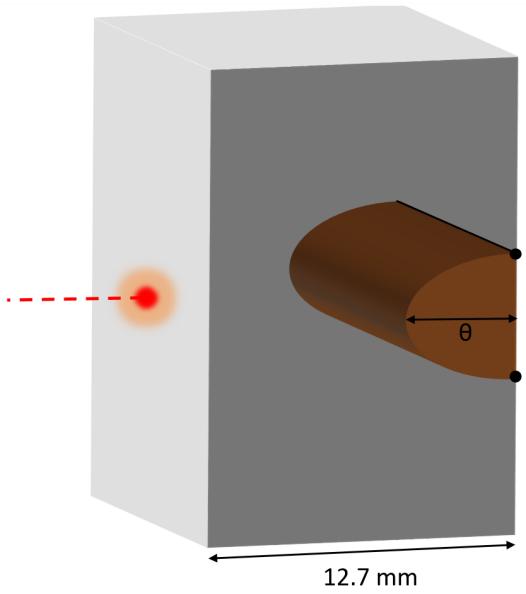

Figure 3.11: Parameterized corrosion pattern within a steel gusset plate. The parabolic corrosion boundary is "anchored" at the points shown in black, and grows out away from the rear boundary. Heat is input to the system on the front face, where the resulting temperature profile is also recorded.

rosion depth, $\hat{\theta}$, and evaluating the probability that the resulting FEM solution gave rise to the observed temperature data after passing through the camera noise model. This is known as the *likelihood*, $p(\mathcal{D}|\hat{\theta})$. Assuming an "uninformative" uniform distribution over θ, that is called a *prior*, the corrosion depth is equally likely to be anywhere within the gusset plate thickness, Bayes' theorem states that the posterior distribution is proportional the likelihood distribution, $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)$, only scaled by a constant. Thus, the problem of estimating the posterior distribution can be reduced to sampling candidate values for θ and evaluating their corresponding likelihoods.

Markov chain Monte Carlo provides a systematic algorithm for producing the desired samples over θ. Starting from an initial guess, θ_0, a new candidate, $\hat{\theta}$, is randomly generated from some small neighborhood of θ_0. The likelihoods

for each value, given the observed data, are calculated through the use of the FEM solver. If the new sample is more likely to have produced the data, then it is set as θ_1. Even if $\hat{\theta}$ was less likely to have produced the observed data, it may be randomly selected as the next sample with probability $p(\mathcal{D}|\hat{\theta})/p(\mathcal{D}|\theta_0)$. Otherwise, θ_0 advances, repeated as the next sample θ_1. This algorithm, known as Metropolis-Hastings sampling [25], is summarized as

$$\theta_{i+1} = \begin{cases} \hat{\theta} & \text{with probability } \min(1, p(\mathcal{D}|\hat{\theta})/p(\mathcal{D}|\theta_i)) \\ \theta_i & \text{otherwise.} \end{cases}$$

Under some technical conditions that are easily verified for our system [41], the sequence $\{\theta_i\}_{i=1,2,3,\ldots}$ is guaranteed to converge to samples from the desired distribution $p(\theta|\mathcal{D})$. The sampling process is summarized in Figure 3.12. The FEM implementation developed in this work is nested within two loops, as the critical stage of computation. Our GPU approach is massively parallelized for this task, and is designed to provide rapid successive solutions with only the transfer of θ_i from the host to the device (i.e. very low communication overhead). Furthermore, if the elemental assembly data is precomputed and loaded onto the device, the entire process can be carried out with minimal memory transfer.

It is customary when using MCMC to discard some number of initial samples, to allow the chain to "burn in" to the posterior distribution, and away from its arbitrary initial value. For these experiments, we perform 200 burn-in iterations, starting from the middle of the prior distribution, and then record the subsequent 2500 samples. The trajectory of the Markov chain and a histogram of the samples are shown in Figure 3.13. It can be seen that the samples fall tightly around the ground truth value of 3.175 mm, denoting a confident solution to the coefficient inverse problem. The mean estimate is 3.16 mm with standard deviation 0.05 mm.

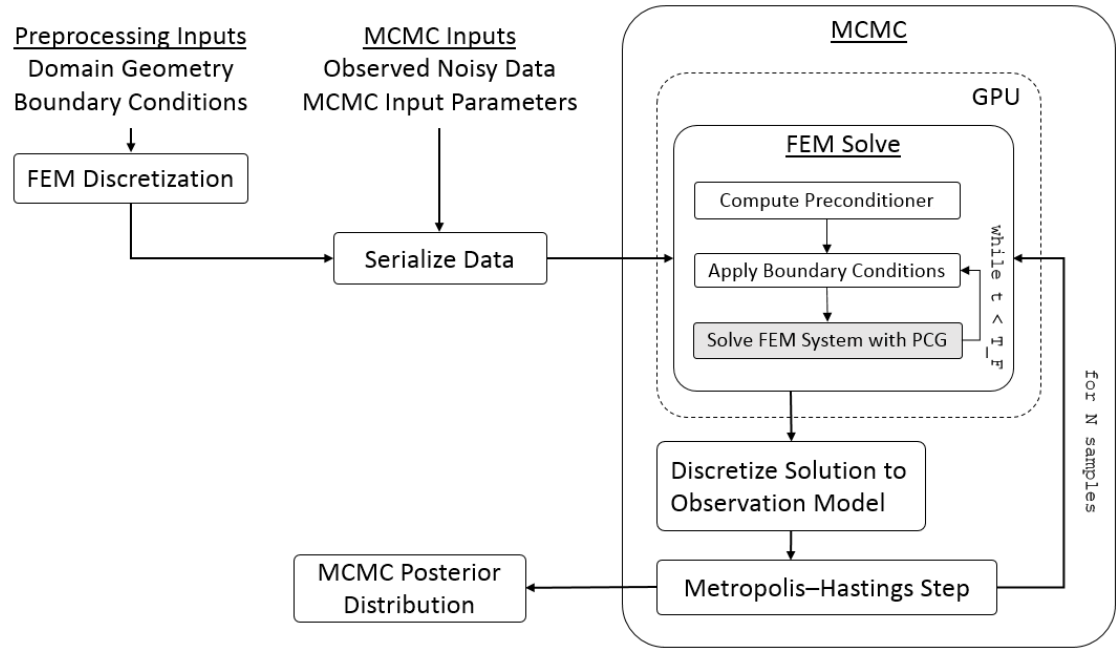

Figure 3.12: Algorithmic flowchart for solving a coefficient inverse problem with MCMC and the heterogeneous computing FEM methods described in this work.

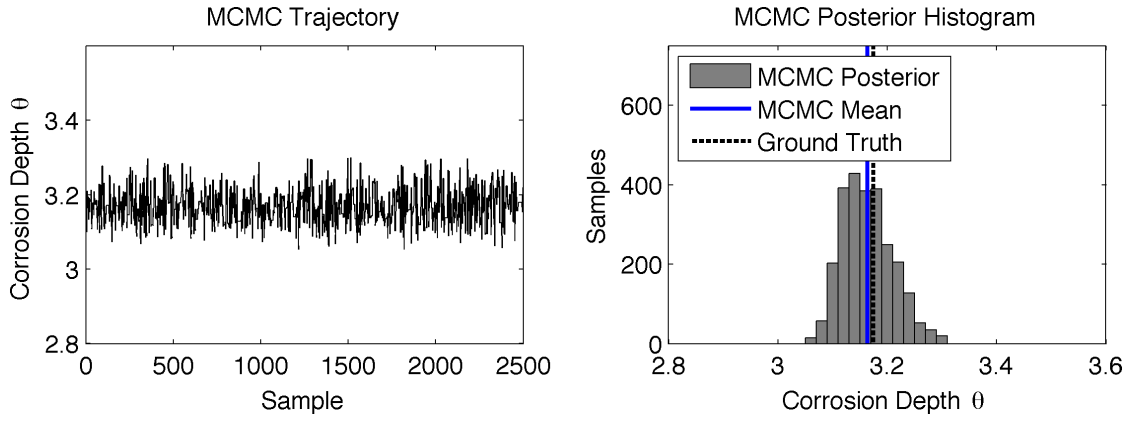

Figure 3.13: (left) Trajectory of the 2500 sample Markov chain. (right) The same samples plotted as a histogram which approximates the distribution $p(\theta|\mathcal{D})$.

The demonstrated sequence of transient heat conduction simulations completed in 9 hours. If the same simulations are attempted using FEniCS, a modern open-source FEM computing platform [44], the system runs out of memory in the assembly process. Computing on a series of smaller problems and extrapolating the run times (ignoring all contributions from just-in-time compilation) yields an expected total time of 142 hours. Using the in-house sparse matrix CPU code that was written specifically to solve this kind of sequence of heat conduction simulation, and discussed in Section 3.5.3, would have taken approximately 38 hours.

3.6 Conclusions

In this paper we described three implementations of a fine-grained assembly-free finite element simulation of heat transfer through heterogeneous media with the use of heterogeneous computing. The methods are motivated in terms of the interpretation of the FEM assembly operator as well as their particular implementation on GPUs. The ease of application of these methods to problems with spatially varying material properties is a direct consequence of the decomposition of the problem into local geometric components, rather than focusing on a global system matrix. Furthermore, successive simulations have low overhead, as mesh generation and computation of elemental assembly matrices can be done in advance. For these reasons, the methods that are developed in this research are particularly well-suited as forward models for coefficient inverse problems and topology optimization problems involving transient heat conduction.

We finally note that the algorithm design methods that we have described are not restricted to the heat equation, and can be used as guidelines for solving other PDEs with spatially varying coefficients. To promote the use and extension of our work, we have used the non-proprietary OpenCL API and made the code for our best-performing implementation (both single and dual GPU versions) publicly available [42].

CHAPTER 4

THERMOGRAPHIC DETECTION AND CHARACTERIZATION OF PITTING CORROSION IN STRUCTURAL COMPONENTS

4.1 Introduction

Of all possible forms of corrosion, pitting corrosion is considered among the most dangerous [57]. It is classified by localized oxidation damage within a metal structure, beginning at its surface, and can be difficult to detect despite posing the threat of failure to an entire engineering system [20, 59]. Pitting corrosion can act as an initiation site for cracks, that may subsequently propagate deeper into the structure [58]. To make matters worse, pitting cavities may become filled with the solid products of corrosion, further confounding the ability to detect them and characterize their internal structure. Protecting against this mode of corrosion is of particular interest when considering steel structures. While normally protected by a thin oxidized layer, steel can undergo a "passivity breakdown," leading to point defects in unpredictable locations which lead to pitting [45]. In the current paper, we propose and use simulations to demonstrate a series of nondestructive testing procedures for the purposes of detecting and characterizing pits of corrosion on visually inaccessible areas of a steel structure within a non-contact context (i.e. the method does not require physical contact with the structure).

The proposed methods in this work are linked though a framework which allows prior knowledge of a problem to be mathematically encoded, along with imperfect experimental measurements, to make optimal decisions in the presence of uncertainty. In the current work, analytical models from the theory of

heat transfer drive rapid, high fidelity simulations, without which the proposed techniques would not be feasible. As an intermediate stage of the pitting corrosion characterization process, these mathematical and computational tools are employed to determine the optimal nondestructive, non-contact inspection context for each individual case of damage that was detected in the initial scan. This links the detection and characterization problems to form a procedure for fully understanding otherwise hidden danger, both early and effectively.

4.1.1 Scope and Organization

This paper is divided into six sections. In Section 4.2, some discussion of earlier related corrosion characterization problems is provided. Next, Section 4.3 covers details of the modelling aspects for our proposed method, both mathematical and computational. Practical considerations regarding our design choices and some of their limitations are also covered. We describe our Bayesian inference procedure in Section 4.4, following a summary of the necessary mathematical tools. The inference procedure is illustrated within a simulated scenario, which is referenced throughout this section and then finalized in Section 4.5, along with interpretations of the findings. Finally, Section 4.6 summarizes the goals and future impact of this research.

4.2 Background and Motivation

Detection and characterization of structural damage due to corrosion is necessary for making informed maintenance and repair decisions. A corrosion at-

tack may be hidden from visual inspection due to its size or location within a large structure. Therefore, other nondestructive testing (NDT) modalities must be employed for investigative purposes. Active infrared (IR) thermography is a modality of NDT by which otherwise hidden properties of a structure are characterized through their interaction with a known thermal input [56]. The properties of interest are often localized damage sites which are too small to view directly, such as nascent-stage cracks [43], or which occur hidden within the structure, as in the case of corrosion [68]. The external, noncontact heat source may come in the form of a flash lamp, continuous wave laser, or frequency modulated laser, as long as its spatial energy flux profile is known by the experimenter, along with salient thermal properties of the structure under evaluation. Variation in the thermal response of the specimen, as measured by an IR camera, may then provide information for the inference of the unknown structural properties. A history of IR technology and the development of pulsed thermal NDT is presented in Ref. [69].

The thermographic characterization of corrosion has been studied within the theory of partial differential equations (PDEs) from two branches of research. In the first, a known domain is taken to have spatially varying material properties, obeying a function that is to be determined from heat information on part of the boundary [30, 52, 65]. This defines a kind of *coefficient inverse problem*. Theoretical results generally take the form of uniqueness of the coefficient function, as well as stability of the function (bounds in a function space norm) to perturbations in the boundary data, using Carleman-type estimates. The strongest such results for the heat equation allows for piecewise smooth coefficient functions [52], while stronger assumptions are required for more non-parabolic PDEs. The second branch of research models corrosion as a vacancy of part of the domain

itself [8]. Thus the problem is in determining the geometry of a portion of the domain boundary from heat measurements taken on a different portion. Again, uniqueness results have been proven, and require weaker assumptions than the previous case [8]. We note that, in the limit of material property values, this second branch of research can be viewed as a special case of the first. The broad class of coefficient inverse problem are expected to be more difficult to solve, given the ill-posed nature of inverse heat problems in general [21].

The two modelling perspectives discussed above persist in current research. Corrosion characterization through coefficient inverse problems have been the focus of numerical studies in recent years [12, 18, 26]. Numerical computation-based study of the missing material setting has also been done [15], with all of the finite element method simulations in the cited literature being restricted to two-dimensional domains. Finally, experimental research has been done in the second case [46, 68].

The current study focuses on corrosion detection and characterization as a coefficient inverse problem in 3D, using a pulsed laser beam as the thermal energy source. It departs from earlier work in 2D [18] by using a heterogeneous computing approach for high fidelity simulation of heat conduction at scales that were previously unfeasible.In addition to the extensions of earlier characterization techniques, we also propose an inspection method for detecting the hidden damage in the first place, and then a framework for generating IR field data that will provide optimal information for the characterization stage of analysis. The experiment optimization stage leads to a practical inspection methodology and is important for generating a useful thermal signal from small or nascent-stage damage with a noisy sensor. It has been shown that a rigorous

treatment of experiment optimization can significantly impact the quality of the final characterization [43], and this stage of analysis is critical for the present 3D coefficient inverse problem with limited boundary data. The proposed inspection process of corrosion detection and characterization employs a Bayesian framework throughout. This provides for the automatic treatment of measurement uncertainty, as well as the propagation of accumulated information about the corrosion from one stage of analysis to the next.

Our corrosion detection and characterization process is summarized as follows, with further details in Section 4.4. First, the laser source is scanned across a steel structure in a line within the field of view of an IR camera trained on the specimen. Video frames of the thermal profile are analyzed in an online setting for statistical departure from some expected response. Wherever anomalies are detected from the scan, an approximation of the response is computed and used as an initialization for the Bayesian optimization stage. In this stage, the locations of interest are subsequently revisited by interrogating the structure with fixed-position laser pulses and acquiring a set of IR image snapshots. If no departure from the expected response is observed on the second pass, that location is ignored. Otherwise, a short sequence of further laser pulses are used to find the position that maximizes the thermal signal, using a Bayesian optimization approach. Determining the optimal position of the laser spot has been found to be the most interesting and useful experimental parameter to investigate in previous research [43]. The IR snapshot that has the greatest thermal signal is then passed to the final characterization stage, in which stochastic numerical simulation is employed to generate probability distributions over parameters that describe a pit of corrosion. In place of actual observed data, a high fidelity heat conduction simulation is used, along with appropriate contamination with

noise, to produce surrogate IR field data. This is a practical necessity for the current study, though we hope to motivate experimental validation of our proposed methods. Lastly, we note that, while we limit the current study to the thermal properties of mild carbon steel and its corrosion products, the methods that we develop can be applied to corrosion (or hidden domain boundary) characterization for any metallic material.

4.3 Problem Description

In this section, the mathematical and computational details of our assumed models are described. We also discuss the design choices that have been made from a practical standpoint.

4.3.1 Mathematical Formulation

The heat equation is a parabolic PDE which describes the flow of heat within a solid medium due to conduction. Within its boundary conditions, heat flux can encode the spatial and temporal addition of heat to the domain of interest, or to specify that no heat transfer occurs, in the case of an insulated boundary.

The current work considers a flat, rectangular, 10 mm thick steel panel, wide enough in the transverse direction so that boundary effects are negligible, 20 mm away from the laser scan line or the fixed laser spot (determined by analytical and computational considerations). The thickness of the panel is chosen to be similar to structures of engineering interest [1] and on the same scale as previous experiments [18, 46, 68]. The corrosion product is assumed to be primar-

ily iron (III) oxide, Fe_2O_3, with salient material properties taken from literature [18]. The relevant material properties for these are *density*, ρ (uncorroded: 0.0076 g/mm^3, corroded: 0.003 g/mm^3), *specific heat*, C (uncorroded: 4.9e8 mm^2/C s^2, corroded: 5.5e8 mm^2/C s^2), and *thermal conductivity*, k, assumed to be constant (uncorroded: 4.3e7 g mm^2/C s^3, corroded: 4e6 g mm^2/C s^3). Our mathematical model is simplified by assuming that the panel under inspection has perfect absorptivity (i.e. behaves as an ideal black body), so that all of the laser energy is converted to heat. We also neglect convection and radiation surface effects, so that the boundaries of the domain are fully insulating. These neglected effects are expected to be small over the short inspection times considered, as supported by numerical considerations, and in the literature [60].

A small pit of corrosion on the back side of the panel is modelled as a radially symmetric 3D Gaussian function, parameterized by four values: the corrosion pit center, (x_c, y_c), penetration depth, d, and width, w. Thus the material property coefficients take their corroded values at point (x, y, z) if

$$z \geq 10 - d \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2w^2}\right).$$

Collectively, these values are gathered into the vector $\vec{\theta} = (x_c, y_c, d, w)$, which will be the damage parameters for inference discussed in Section 4.4.5. With different values of these damage parameters, the range of expected profiles of pitting corrosion can be modelled, both sharp and wide [57]. The mathematical description of the domain, as well as an illustration of the experimental setup are shown in Figure 4.1.

The energy input for our experiments is furnished by a 100 W laser beam with 2D Gaussian energy density profile [3], directed at a point $\vec{x}_l(t) =$

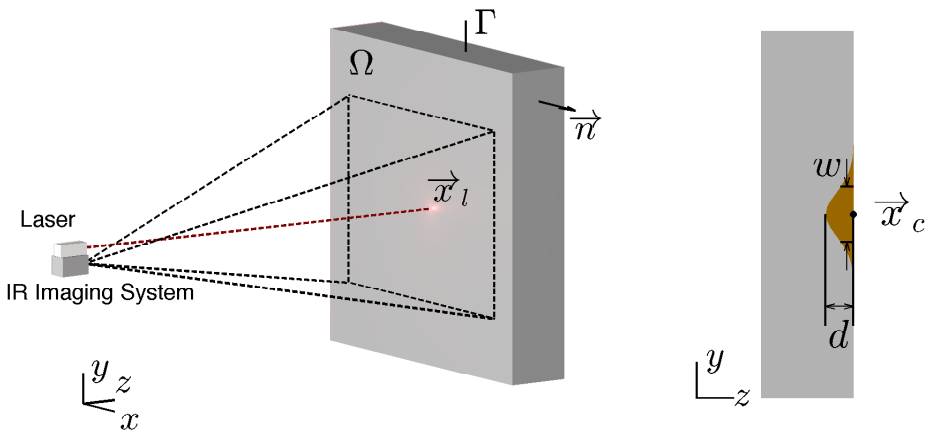

Figure 4.1: Diagram of the problem domain, labeled according to the mathematical formulation. (left) A 3D steel panel under inspection, with (right) a 2D cross-section into its depth through the center of a pit of corrosion.

$(x_l(t), y_l(t), 10)$ on the front surface of the panel as:

$$f(\vec{x}, t) = \begin{cases} \frac{2P}{\omega^2} \exp\left(-\frac{2\left((x-x_l)^2+(y-y_l)^2\right)}{\omega^2} \right) & \text{if } z = 0, \\ 0 & \text{otherwise,} \end{cases}$$

where P is the *laser power* and ω is the *beam width* (the radius at which the intensity has dropped to $1/e^2$ of its peak value). In the case of the corrosion detection procedure, the laser is scanned in a line across the steel panel. For the subsequent characterization stages, the laser spot position is stationary in time. This particular mode of thermal energy input is chosen, as opposed to flash lamps, or some other source, so that these two use cases can be implemented with the same set of inspection equipment. By adopting the proposed procedures, a large structure can be examined broadly, and then in detail as needed, with larger standoff distances than are feasible when using flash lamps.

The location of the laser spot, $\vec{x}_l(t)$, during detailed inspection is the primary experimental parameter that will be optimized in this work, with methods described in Section 4.4.4. The pulse duration, τ_h, and subsequent cooling time, τ, are also important design choices, and have been studied for similar corrosion characterization problems by Vavilov et al., to balance sensitivity of the measurement to the back of the panel with 3D diffusion effects which dissipate the signal [68]. We take the suggested value of τ as 70% of the "end time" of the process, using the temperatures below the laser spot, throughout the thickness of the panel, to determine a steady state, rather than using the Fourier number of the domain. This is done because energy is not uniformly deposited into the front of the structure, and it would take much longer for the temperature to reach equilibrium in the transverse directions (away from the laser spot center). Experimental determination of this value, using a 10% cutoff threshold for the end process limit [68], yields the value $\tau = 3.3$ seconds. We set $\tau_h = 1$ second so that the panel temperature increases enough for a useful measurement, and have ensured that the pulse duration is short enough that perturbing it does not significantly affect the end process time.

The resulting PDE boundary value problem that models our experiments is given in strong form by

$$\begin{cases} \rho(\vec{x})C(\vec{x})\frac{\partial T(\vec{x},t)}{\partial t} - \nabla \cdot (k(\vec{x})\nabla T(\vec{x},t)) = 0 & \text{in } \Omega \times (0,\infty), \\ k(\vec{x})\frac{\partial T(\vec{x},t)}{\partial \vec{n}} = f(\vec{x},t) & \text{in } \Gamma \times (0,\tau_h], \\ k(\vec{x})\frac{\partial T(\vec{x},t)}{\partial \vec{n}} = 0 & \text{in } \Gamma \times (\tau_h,\tau_h+\tau], \\ T(\vec{x},t) = T_{\text{ambient}} & \text{on } \Omega \times \{t=0\}, \end{cases} \tag{4.1}$$

where $\vec{n}$ denotes the outward normal vector to the domain and T_{ambient} is the initial, constant temperature of 25 °C. The domain Ω is the 3D region of the

panel around the spot of inspection with boundary Γ, as depicted in Figure 4.1.

4.3.2 Forward Modelling

The characterization of a pit of corrosion from thermal data can be posed as an *inverse problem*. One class of inverse problems seeks a solution yielding information concerning underlying parameters instantiating a system, by leveraging that system's observed behavior to propose plausible model instances of the actual system that match the observations. This is in contrast to a forward problem, which is the prediction of a system's behavior, based on its known physical properties and parameters. In our case, the forward problem is finding the thermal response of a corroded steel panel with a corrosion pit with known size, shape, and location. This can be solved using the heat equation and the *finite element method* (FEM), as a means of discretizing and numerically treating Equation (4.1).

In this work, we solve the forward problem as a weak form [19] using an in-house developed heterogeneous computational framework, based on the finite element method for heat conduction simulation over a heterogeneous domain. The software uses an assembly-free finite element method whose resulting linear system is treated in a conjugate gradient solution on a graphics processing unit (GPU) with Jacobi preconditioning [63]. High-fidelity simulations are performed more rapidly than with standard FEM programs. Furthermore, there is virtually no overhead in providing successive simulations that respect changes in the spatial corrosion parameterization, since the FEM system matrix is never explicitly assembled. These properties make the chosen forward model solu-

tion method ideal for our current use case in which many, possibly thousands, of forward simulations are necessary to provide a solution to the corresponding inverse problem.

The heterogeneous computing framework solves the heat equation PDE over a fixed grid linear tetrahedral FEM mesh. Time discretization is done with the Crank-Nicolson method [44]. A spatio-temporal convergence study is carried out to find the appropriate spatial mesh refinement, as well as needed time step size. Convergence is confirmed across successive refinements as well as with comparison to the analytical solution in the case of a laminar composite material structure. We use the criterion that differences in successive refinements are not distinguishable when the solution is passed through the imaging system that will be described later herein. Convergence is observed for time steps of 0.01 seconds and a spatial mesh of first-order Lagrangian tetrahedra comprising approximately 1.2 million degrees of freedom within a structured mesh whose element sizes remain uniform.

The finite element model furnishes an approximation to the heat response on a continuum. From this, realistic surrogate experimental data are generated, as if they were recorded by an actual thermal imaging system, through the following process. Field variable output from the FEM mesh nodes are interpolated onto a finer rectangular grid, then integrated over the area which would be captured by each pixel of the hypothetical imaging system (to simulate the integration occurring within each pixel of the assumed microbolometer array). Next, independent, identically distributed (i.i.d.) Gaussian measurement noise is added to each pixel reading, in a manner that is consistent with the noise-equivalent temperature difference (NETD) of our assumed microbolometer sys-

tem. Finally, the data are rounded to be consistent with specified resolution, ΔT, that is associated with the quantization assumed in our image capture. We assume a research-grade camera (A655sc from FLIR Systems, Inc) as the basis for our modeled imaging system, which has a NETD of 0.03 °C, a standard temperature range from -20 °C to 120 °C, frame rate of 50 Hz, and 16-bit data representation, for a thermal resolution of $\Delta T = 0.003$ °C. The spatial resolution (640 $\times$ 480) and angle of view of this camera ($15°$) are determined to give a pixel size of 1 mm $\times$ 1 mm with a standoff distance between the camera and the specimen of 2.4 m.

4.3.3 Practical Considerations

We have made a number of simplifying assumptions throughout the formulation of this problem for the sake of modelling and computational tractability. Here we will discuss their justifications and limitations. We model one isolated area of corrosion with a radially symmetric Gaussian profile. Our proposed detection method allows for several such pits as long as they are far enough away that their thermal responses do not interfere. In reality, the threat of nearby instances of corrosion pits is decreased, since they must share the available cathodic area, and hence grow more slowly [57]. Additionally, our characterization methods can be extended to model more complex corrosion geometries. For example, a mixture of Gaussians with non-isometric eccentricity is an expressive function basis, and would require minor modifications to the current theory to model. Next, we assume that the additive Gaussian noise model will drive the stochasticity in our inverse problem solution, while the quantization error due to the bit depth of the A/D converter in our assumed CCD is omitted

from the forthcoming probabilistic models. It was shown in Ref. [43] that for values of NETD/ΔT greater than 2, measurement noise is the dominant factor over quantization error. The value for our assumed thermal imaging system model is 10, so we neglect the quantization in Section 4.4. The effect is maintained in the contamination process for surrogate field data. The sensor noise is assumed to have no spatiotemporal correlation, or scale with signal strength for analytical tractability [2]. The last assumption of Gaussianity that we have made is for the laser energy density. This choice, rather than, say, uniform energy over a circular spot, is well supported [55].

Further assumptions have been made for the sake of convenience. Full absorbtivity of the laser energy as heat is not realistic, but can be corrected for with a proper scaling of the power coefficient. We have taken there to be no thermal contact resistance at the interface between steel and the corrosion products, which would not be the case for any imperfect surface contact between the two. The details of the thermal imaging lens and standoff distance were chosen to give a pixel area of 1 mm^2. These choices are flexible, and can be adjusted for higher image resolution or a greater standoff distance, as required for a particular application. Provided that the associated computational model is properly modified, our methods can still be applied. As another assumption, we have limited to a binary choice of material: either ideal steel or corrosion product, with values for material properties taken from literature. The thermal properties of the material could be treated as unknown, and then inferred with the current methods. A smooth transition region between the two materials could also be included with additional parameterization. Both of these generalizations would give more realistic models at the expense of an increase in the dimension of our inference space, and are left to future work.

4.4 Bayesian Infernce Methodology

We now propose a corrosion detection and characterization methodology within the previously stated model domain. The underpinning mathematical theory for for Bayesian analysis will be described first. Then the Bayesian inference procedure will be introduced through an example simulation. The inference pipeline of the proposed process is summarized in Figure 4.2. Notation that is used throughout this paper is summarized in Table 4.1.

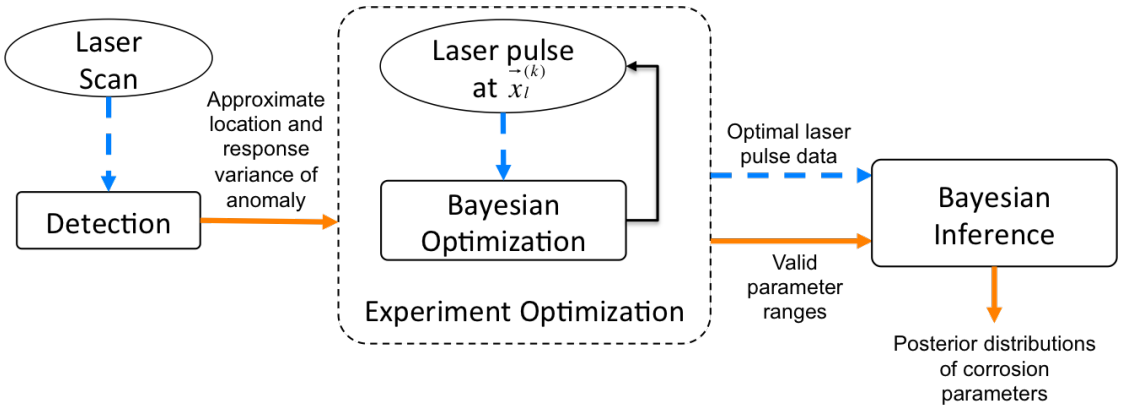

Figure 4.2: Inference pipeline of the proposed process. Dashed blue lines denote IR measurement data (simulated or experimental), while solid orange lines indicate learned data regarding the corrosion pit parameterization.

4.4.1 Bayesian Inference Background

The goal of this work is to solve a stochastic inverse problem to uncover information characterizing a small pit of corrosion with parameterization values $\vec{\theta} = (x_c, y_c, d, w)$ by leveraging data from a noisy thermal camera measurement, $\mathcal{D}$. For an image with n pixels, the observation model put forth in Section 4.3.2

Symbol	Meaning	Values Taken
Continuous Mathematical Formulation		
T	Exact solution to heat equation PDE	Real function of $\vec{x}$ and t
$\vec{x} = (x, y, z), t$	Space and time variables	Real numbers, positive for time
x_l, y_l, P, ω	Laser parameters: within-plane spot location, power, and beam width	Real numbers, positive for power and beam width
$\vec{\theta} = (x_c, y_c, d, w)$	Corrosion pit parameters: within-plane location, penetration depth, and width	Real numbers, positive for depth and width
ρ, C, k	Material density, specific heat, and conductivity	Positive real numbers, taken from literature
τ_h, τ	Optimal laser pulse and subsequent cooling times for optimal inspection	Positive real numbers, based on recommended values
$\Omega, \Gamma, \vec{n}$	PDE domain, boundary, and outward normal vector	3D set, 2D surface, and 3D vector
Discrete Observation Model		
$\mathcal{D}$	Experimental data	Set of real numbers
F	Ideal pixel measurements, discretized version of T	Real function of $\vec{x}_l$ and $\vec{\theta}$
n	Number of pixels in data	Positive integer
NETD, ΔT, frame rate, angle of view	Assumed IR camera specifications	Positive real numbers, taken from A655sc data sheet
Y	Noisy pixel measurements	Real valued multivariate random variable
$\hat{Y}$, $\hat{F}$	Deviation of Y or F from the ideal response on an uncorroded domain	Real numbers
ε, σ_{NETD}	Random noise used in the mathematical observation model and its standard deviation	Real random variable and postive real number from specified NETD
Bayesian Optimization and Inference		
a_1, a_2	Hyperparameters for squared exponential covariance function	Positive real numbers
G	Optimization objective function	Real valued random function of $\vec{x}_l$
g	Gaussian process	Real random function
$\mathbf{K}$, $\vec{k}$	Covariance arrays from pairwise evaluations of κ	Real valued matrices
$S^{(m)}$	Cumulative deviation over m frames	Real numbers
μ, κ	Mean and covariance function for a GP	Real valued functions
μ^+, $\mathcal{D}^+$, $\vec{x}_l^+$	Optimal values of GP posterior mean, experiment data, and optimization index	Same as non-optimal values
π	parameter index for Metropolis within Gibbs proposal	$\{1, 2, 3, 4\}$

Table 4.1: Notation that is used throughout this work. Within the corrosion detection laser scan, some values are time-dependent.

is summarized as

$$\begin{aligned} \mathcal{D} &= \{Y_i\}_{i=1}^{n}, \\ Y_i|\vec{x}_l,\, \vec{\theta} &= F_i(\vec{x}_l, \vec{\theta}) + \varepsilon_i, \\ \varepsilon_i &\sim \mathcal{N}(0, \sigma^2_{\text{NETD}}), \end{aligned} \tag{4.2}$$

where $F_i(\vec{x}_l, \vec{\theta})$ is the idealized noise-free value of the thermal response averaged over the pixel area. Mathematically, the noise induces stochasticity in the inference, and our task is to estimate the probability distribution $p(\vec{\theta}|\mathcal{D},\, \vec{x}_l)$. Bayes' theorem provides a path to this distribution, called the *posterior*, through knowledge of other distributions [41]:

$$p(\vec{\theta}|\mathcal{D},\, \vec{x}_l) \propto p(\vec{\theta})p(\mathcal{D}|\vec{\theta},\, \vec{x}_l). \tag{4.3}$$

The right-hand size of Equation (4.3) comprises the *prior* distribution $p(\vec{\theta})$, which encodes all previously known information of the parameters, and *likelihood* distribution $p(\mathcal{D}|\vec{\theta},\, \vec{x}_l)$, which can be evaluated with a forward simulation that is instantiated using corrosion parameters $\vec{\theta}$. The likelihood that parameters $\vec{\theta}$ gave rise to observed data, $\mathcal{D}$ through Equation (4.2), is the probability that the deviation from the ideal response is due to i.i.d. Gaussian noise

$$\begin{aligned} p(\mathcal{D}|\vec{\theta},\, \vec{x}_l) &= \prod_{i=1}^{n} \mathcal{N}\left(Y_i - F_i(\vec{x}_l, \vec{\theta}); 0, \sigma^2_{\text{NETD}}\right) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2_{\text{NETD}}}} \exp\left(-\frac{\left(Y_i - F_i(\vec{x}_l, \vec{\theta})\right)^2}{2\sigma^2_{\text{NETD}}}\right) \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2_{\text{NETD}}}}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}\left(Y_i - F_i(\vec{x}_l, \vec{\theta})\right)^2}{2\sigma^2_{\text{NETD}}}\right). \end{aligned} \tag{4.4}$$

We are interested in the relative likelihoods coming from many different parameter realizations, which may result in very small differences in the thermal response of the specimen at its front face. Any information that is to be used for

inference will be contained within the difference between the corroded panel's thermal response and the corresponding response of a specimen with no damage, $\hat{F}(\vec{x}_l, \vec{\theta}) = F(\vec{x}_l, \vec{\theta}) - F(\vec{x}_l, 0)$. For corrosion detection to be possible at all, we need this difference to exceed the sensor noise floor, or else $\hat{Y} = Y - F(\vec{x}_l, 0)$ will be dominated by the noise. Similarly, the theoretical limitation of any characterization procedure is governed by the relative responses from different parameterizations, compared with the scale of σ_{NETD}. The experimental optimization methodology that is developed, and that is aimed at uncovering a useful inspection modality, in this section is motivated by maximizing the useful inference information, $\mathcal{D}$, in the presence of noise.

The flexibility to mathematically incorporate prior knowledge of the parameter vector to be inferred, $p(\vec{\theta})$, is a strength of Bayesian inference. Although in our case, we begin with no such knowledge, having observed no evidence of damage, we will accumulate and refine information about $\vec{\theta}$ through each stage of the following procedure. In the subsequent sections, we will expand on the way that $p(\vec{\theta})$ is updated from the detection process, to experiment optimization, and then to the final inference stage, as illustrated in Figure 4.2.

4.4.2 Gaussian Process Background

Section 4.4.4 is based on the theory of Gaussian processes (GPs), a useful idea in machine learning for placing a probability distribution over a continuous function [54]. Here we briefly cover the background of GP theory as needed for our corrosion characterization methodology. A GP $g : \Omega \to \mathbb{R}$ is a random function with the property that any finite collection of its values are related to one an-

other through a multivariate Gaussian joint distribution. Just as a multivariate Gaussian random variable can be fully specified by its mean and covariance matrix, the behvior of a continuous Gaussian process is similarly defined through its mean function $\mu(\vec{x})$ and covariance kernel function $\kappa(\vec{x}, \vec{x}')$. We write the continuous function $g(\vec{x}) \sim \mathcal{GP}(\mu(\vec{x}), \kappa(\vec{x}, \vec{x}'))$ if, for any set of points $\{\vec{x}_i\}_{i=1}^N$, the function values satisfy $\vec{g} = \{g(\vec{x}_i)\}_{i=1}^N \sim \mathcal{N}(\vec{\mu}, \mathbf{K})$, with $\vec{\mu}_i = \mu(\vec{x}_i)$ and

$$\mathbf{K} = \begin{bmatrix} \kappa(\vec{x}_1, \vec{x}_1) & \cdots & \kappa(\vec{x}_1, \vec{x}_N) \\ \vdots & \ddots & \vdots \\ \kappa(\vec{x}_N, \vec{x}_1) & \cdots & \kappa(\vec{x}_N, \vec{x}_N) \end{bmatrix}.$$

Gaussian processes can enforce rich classes of behavior over the function model, such as varying amounts of smoothness or periodicity, depending on the choice of mean and covariance function [54]. These properties, along with an analytic form for the predictive distribution at an unknown point, $\vec{x}_*$, make GPs powerful tools for arriving at decisions in the presence of uncertainty. Suppose N observations, $\vec{y}$, of an unknown function have been made at (possibly) different points. Then a GP regression can be performed to model these observations, as well as induce a Gaussian distribution at the unobserved point $g(\vec{x}_*) \sim \mathcal{N}(\mu_*, \sigma_*^2)$, where [54]

$$\begin{aligned} \mu_* &= \vec{k}^T \mathbf{K}^{-1}(\vec{y} - \mu(\vec{x}_*)) + \mu(\vec{x}_*), \\ \sigma_*^2 &= \kappa(\vec{x}_*, \vec{x}_*) + \vec{k}^T \mathbf{K}^{-1} \vec{k}, \\ \vec{k} &= \begin{bmatrix} \kappa(\vec{x}_*, \vec{x}_1) & \cdots & \kappa(\vec{x}_*, \vec{x}_N) \end{bmatrix}^T. \end{aligned} \tag{4.5}$$

We leverage the capabilities introduced here: encoding prior knowledge of an unknown function and its qualitative behavior through mean and covariance functions, and retrieving probability distributions of the function's values away from observed data, to incorporate Gaussian processes into our proposed Bayesian inference procedure.

4.4.3 Detection Procedure

The proposed procedure for detecting damage within a steel panel is based on a quantification of the probability that a sequence of thermal images resulted from a random noise process, or else has some anomalous localized bias. Data are taken from the assumed IR imaging device in the form of a video with m frames as the laser spot is scanned in a line across the field of view. We simulate scans with the linear laser motion at 0.1 m/s using the the forward model of heat conduction and realistic camera measurements, both described in Section 4.3.2, considering situations with and without corrosion. The differences between an observed measurement and the ideal case $F(\vec{x}_l, 0)$ might be due either to sensor noise, or an actual departure in the thermal response due to the presence of a corrosion pit. We aim to separate these two cases in a principled way. Figure 4.3 illustrates the corrosion detection method partway through the process, with a pit of corrosion included with $\vec{\theta}_{\text{true}} = (0, 4, 4, 2)$. Figure 4.3(a) shows a realization of the resulting response $Y^{(85)}$ after 1.7 seconds, and Figure 4.3(b) shows $\hat{Y}^{(85)}$, having subtracted the corrosion-free and noise-free response, $F(\vec{x}_l, 0)$. No signal is discernible at this point.

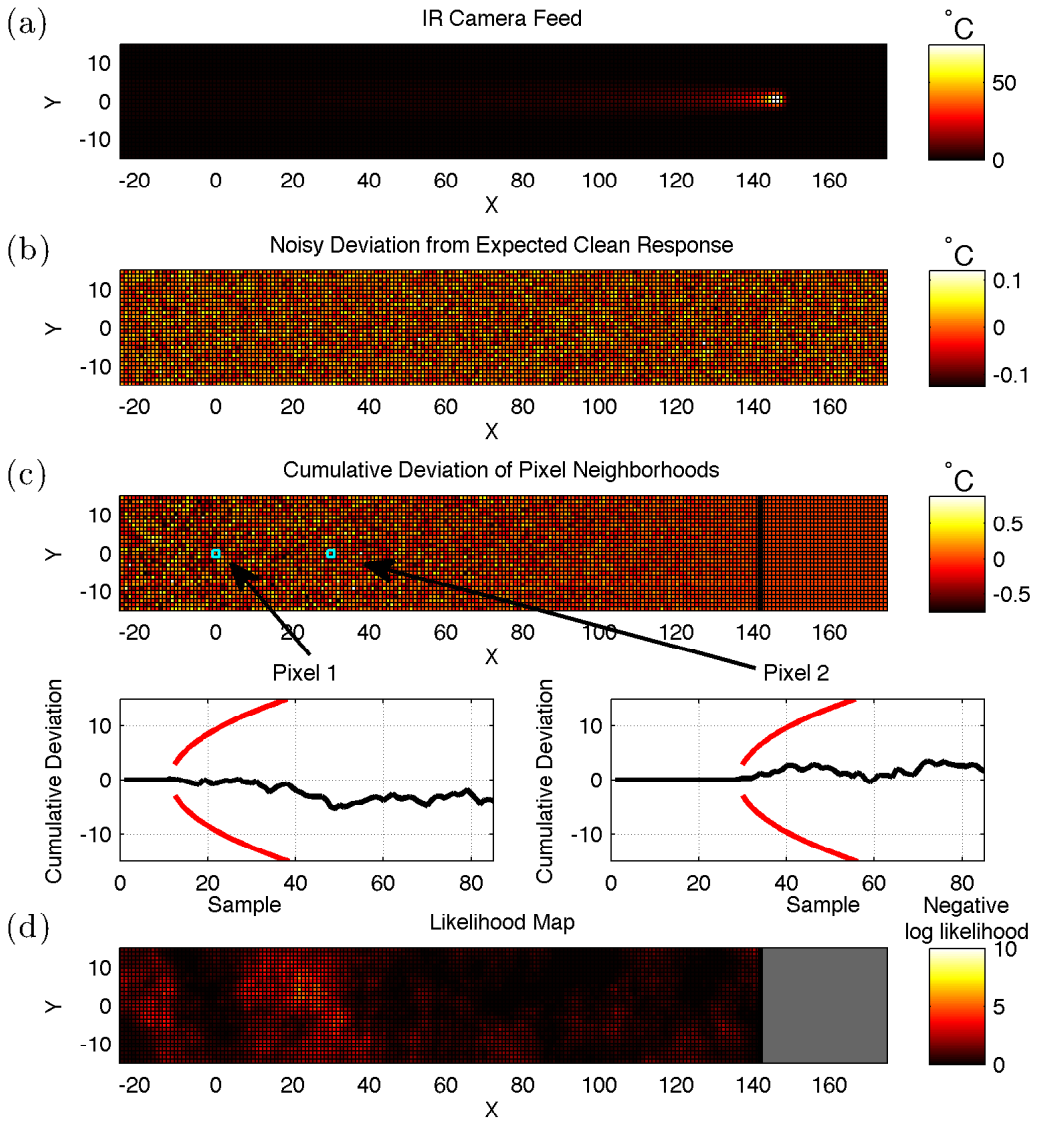

Figure 4.3: Corrosion detection procedure partly through a scan. (a) Simulated pixelated IR response to laser scan and (b) $\hat{Y}^{(85)}$. (c) The cumulative sum of deviations after 1.7 seconds and trailing the laser scan, denoted by the vertical black line. Two pixels are emphasized, with trajectories of their neighborhood cumulative deviations. (f) The negative log likelihood that the measurements from each pixel and its neighbors are the result of a Gaussian random walk.

By adding together measured deviations across $m(i)$ successive frames for pixel i, we define a random walk at each pixel, $S_i^{(m(i))} = \sum_{j=1}^{m}(i)\hat{Y}_i^{(j)}$. The process of accumulation is delayed until the laser has swept across a pixel's horizontal location, plus 0.05 seconds (five pixels trailing the laser center), since no significant thermal variation is expected up to that point, even if the specimen is damaged. In the case that no damage is present in the specimen, $S_i^{(m(i))}$ is a simple random walk with Gaussian distributed step sizes, which has a simple analytical distribution itself, $S_i^{(m(i))} \sim \mathcal{N}(0, m(i)\sigma^2_{\text{NETD}})$. This fact gives a way for us to compute the likelihood that a particular sequence of measurements correspond to an undamaged specimen. Figure 4.3(c) shows the magnitude of $S^{(m)}$ for each pixel. The trajectories of $S^{(m)}$ for two pixels are also plotted, along with a vertical threshold (in red) that corresponds to $\exp(-20)$ of the total probability density for the simple random walk defined by an undamaged specimen. This is the likelihood threshold which we use in flagging a location to have exhibited anomalous behavior (under our assumptions, occurring naturally with probability $\exp(-20) \approx 2 \times 10^{-9}$), thus to be revisited for further investigation. For the final stage of detection, we additionally convolve the cumulative deviations with a circular smoothing kernel of radius 20 pixels, to attenuate noise from the measurements, while emphasizing regions with spatial correlation. The last panel in Figure 4.3 demonstrates the smoothed spatial likelihood map. Regions with higher and lower likelihood of resulting from a Gaussian simple random walk can be seen, but nothing has been flagged as anomalous at this point.

We continue the simulated example detection process until an anomaly is flagged, with the likelihood map at that point shown in Figure 4.4. The deviation trajectory of the flagged pixel is seen to cross the likelihood threshold 4.3 seconds after the laser spot scanned over the pixel. The detection process does

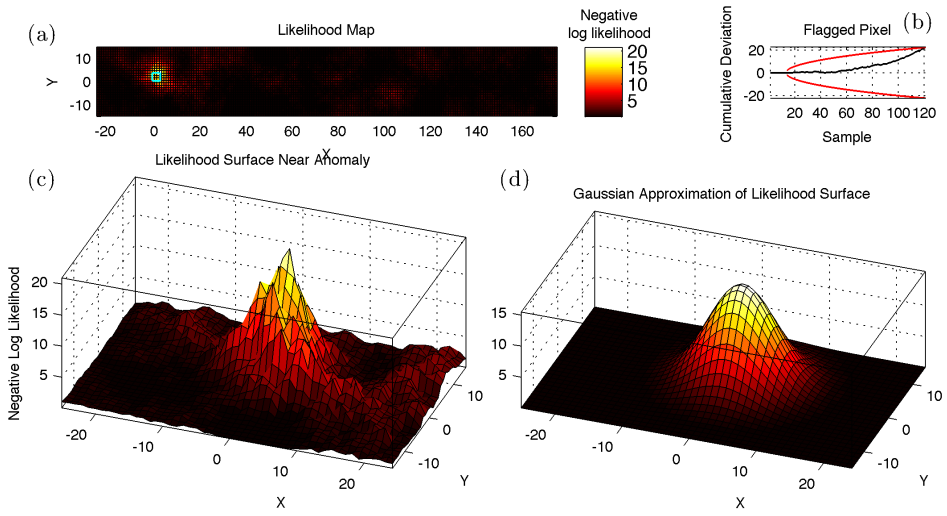

Figure 4.4: Output from a corrosion detection scan after an anomaly is flagged. (a) The likelihood map showing a bright region that has exhibited an abnormal and spatially correlated thermal response, and (b) the cumulative deviation trajectory of the flagged pixel. (c) A 3D view of the likelihood map to compare with (d) the Gaussian surface fit that best estimates it.

not need to be terminated here, so that many anomalies can be detected in a single pass. A 2D Gaussian function can be fitted to the likelihood map at regions of the scan that are flagged, giving a good approximation and allowing the scan results to be encoded into a useful form for later analysis. We use such an approximation to create a prior distribution over the location of a pit of corrosion for the next stage in the characterization process.

Performing simulations of our damage detection process for different sizes and locations of corrosion, we observe some bias in the flagged location, $\vec{x}_{\text{anomoly}}$, towards the laser scan line. For example, the center of the Gaussian approximation with $\vec{\theta}_{\text{true}}$ above is $\vec{x}_{\text{anomoly}} = (1.37, 2.55)$, about half as far from the laser scan line as the corrosion pit. We posit that the bias is due to the fact that heat

is radiating out from the scan line, so that the nearer side of corrosion will interact with and disrupt the response first. The systematic bias reinforces that, while detection with a laser scan can be useful for covering a large surface, it is not necessarily suitable for immediate characterization. We have also observed that the ability for damage to be detected in this way depends on the severity of the corrosion as well as the distance from the laser line. A pit of corrosion only 1 mm deep can be detected 12 mm away from the laser scan line, but not 30 mm away. Hence, the vertical distance between of a series of scans should be governed by the desired sensitivity to the worst case of damage that will be considered, with finer scans for higher sensitivity.

4.4.4 Bayesian Experiment Optimization

Having an estimate for the location of a pit of corrosion, we next seek an optimal experiment for detailed characterization analysis. Bayesian optimization (BO) is an example of a surrogate optimization method, in which a function to be minimized is approximated by a different function which is simpler to evaluate. For Bayesian optimization, a Gaussian process prior is placed over the objective function. In addition to the expressiveness that can be specified with GPs through the covariance function, there are also strong analytical tools for making optimal decisions in the presence of noise [54, 5, 22]. These tools come with a computational burden of evaluating Equation (4.5) on a fine mesh of candidate points. Thus the proper use case for Bayesian optimization is one for which the objective function is expensive to evaluate, or one for which an optimum is desired with as few evaluations as possible [22]. In the case of our current thermographic inverse problem, we consider measurements in the form

of a noisy IR image, after a laser pulse, and following a short period of thermal evolution. We aim to find the laser spot location, $\vec{x}_l$, that will yield the most informative posterior distribution $p(\vec{\theta}|\mathcal{D}, \vec{x}_l)$. Since we want to maximize the information above the noise floor, our objective function $G(\vec{x}_l)$ is chosen to be the sum over all pixels of the differences between the measurement and the expected response of an undamaged structure. The absolute value of $\hat{Y}$ is not taken since deviation due to the thermal insulation by corrosion products will be positive, while deviation due to noise will be positive and negative, and hence cancel to some extent. In addition, the choice to not maximize $\hat{Y}$ in, say, the L_2 norm is made to be consistent with the previous section. The optimization problem is

$$\max_{\vec{x}_l} G(\vec{x}_l) = \max_{\vec{x}_l} \sum_{i=1}^{n} \hat{Y}_i|\vec{x}_l = \max_{\vec{x}_l} \sum_{i=1}^{n} \left(Y_i - F(\vec{x}_l, 0)\right). \tag{4.6}$$

Using this methodology, we must wait for the sample to cool down to a uniform ambient temperature before probing at another laser target. We therefore have an ideal scenario for Bayesian optimization: we can encode prior knowledge of a corrosion pit location, as well as expected qualitative behavior of the objective function through GP mean and covariance functions, respectively. Furthermore, there is a cooling time required between evaluations of the objective function by heating the specimen, so that the necessary computation for determining future laser spot locations can be performed without affecting the total experiment duration.

We use the location and standard deviation of the Gaussian approximation to the detection likelihood surface (Figure 4.4(d)) as the GP mean function. It is expected that the total thermal response will be high when the laser spot aligns with the deepest point in the corrosion pit, $\vec{x}_l = \vec{x}_c$, and that the two measurements of thermal disruption (negative log likelihood and summed signal devia-

tion) will decay on the same scale, since they are based in the amount of themal disruption from its expectation as the laser spot moves away from the corrosion pit. In our proposed approach, we use the isometric *squared exponential* covariance function with i.i.d. additive noise [54],

$$\kappa(\vec{x},\vec{x}') = \kappa(\delta) = a_1 \exp\left(-\frac{\delta^2}{2a_2^2}\right) + \begin{cases} \sigma^2_{\text{NETD}} n & \delta = 0 \\ 0 & \delta \neq 0 \end{cases}, \qquad \delta = |\vec{x} - \vec{x}'|, \quad (4.7)$$

since the noiseless thermal responses should be smooth and should not exhibit directional dependence. The covariance hyperparameters, a_1, and (especially) a_2, can dramatically affect the behavior of the GP, and are therefore informed by evaluating our objective function over a coarse mesh with several hypothetical corrosion profiles. Figure 4.5 shows four such sets of evaluations, as well as the mean of a GP that is fit to these data through the use of standard hyperparameter learning methods [54]. The corrosion parameterizations, as well as their corresponding optimal length scales are displayed. We fix $a_2 = 13$ based on this analysis, to hedge against the worst case of a small pit of corrosion that does not result in much thermal disruption. We adaptively set a_1 to be the response of the first experiment, $G(\vec{x}_{\text{anomoly}})$. Having computed a GP regression fit with these hyperparameters, and any observed function values from earlier experiments, the next laser spot to query is chosen by maximizing an *acquisition function*. We choose the popular *expected improvement* function [5]

$$\begin{aligned} \text{EI}(\vec{x}_*) &= (\mu_* - \mu^+)\Phi(Z_*) + \sigma_*\phi(Z_*), \\ Z_* &= \frac{\mu_* - \mu^+}{\sigma_*}, \\ \mu^+ &= \max_{\vec{x}\in X} \mu(\vec{x}), \end{aligned} \quad (4.8)$$

evaluated on $\vec{x}_* \in X$, a fine mesh of 20,000 points. The GP specification functions here are updated with all previous data. Here Φ and ϕ denote the cumulative and probability density functions of the standard normal distribution,

respectively. This evaluation takes into account high values of the GP mean close to observed data, as well as the potential for improvement offered by a high GP variance further away from previously evaluated points.

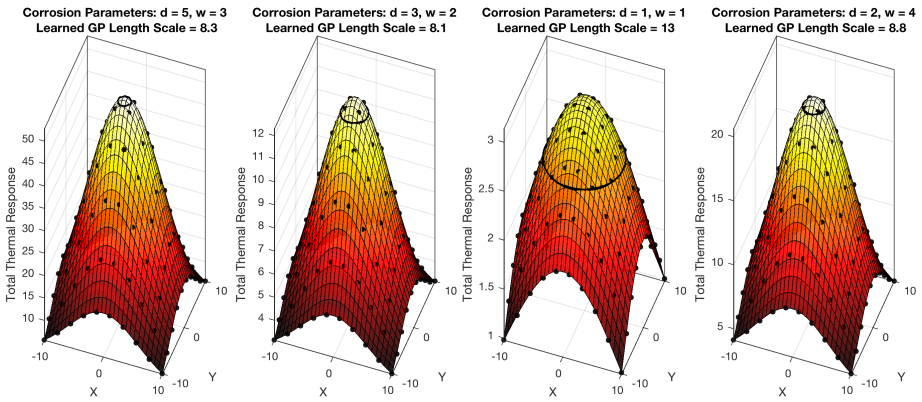

Figure 4.5: Pointwise evaluations of the experiment optimization objective function $G(\vec{x}_l)$ for four different corrosion pit parameterizations (black dots). Gaussian process regression with hyperparameter learning via maximum marginal likelihood is performed for each set of data. The GP mean function and learned length scale a_2 are shown. Additionally, a black ring around each surface corresponds to a level set for an assumed noise of $\sigma_{\text{NETD}}\sqrt{n}$ away from the peak.

We use the BayesOpt lightweight implementation of Bayesian optimization [22] for the task of finding an optimal laser spot location/thermal response data, as follows. The first measurement is taken with the laser spot directed at the center of the mean function: corresponding to the peak likelihood location found by the first scan, $\vec{x}_l^{(1)} = \vec{x}_{\text{anomoly}}$. The second laser spot $\vec{x}_l^{(2)}$ is chosen randomly nearby. Then from these two points of data, Gaussian process regression is performed and the expected improvement is computed at each point in the fine mesh X. The point with the greatest expected improvement is selected as $\vec{x}_l^{(3)}$, the laser spot for the third experiment. We have found that a corrosion pit $\vec{\theta}(0, 12, 1, 4)$ can be found to within 3 mm after 7 iterations in this way. Figure

4.6 illustrates representative Bayesian optimization iterations resulting from the corrosion pit and initial detection scan scenario in Figures 4.3 and 4.4. Contours representing the GP mean conditioned on oberseved data, and the expected improvement surface are shown. Observe that the proposed locations are not necessarily chosen to maximize the GP mean function.

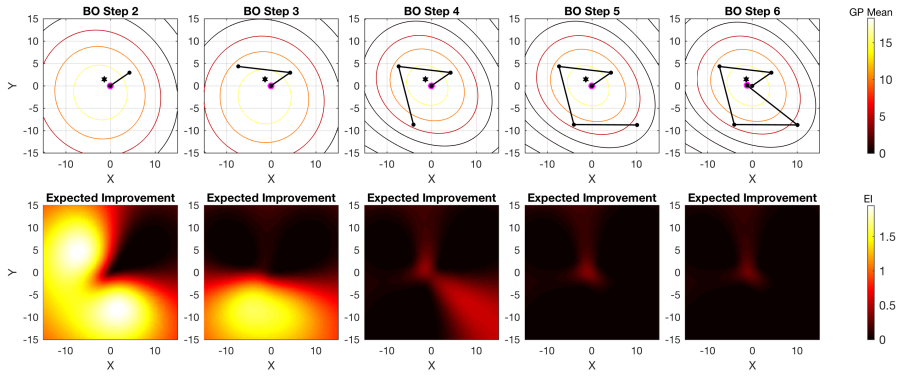

Figure 4.6: Five iterations of Bayesian optimization beginning from the Gaussian fit to the likelihood map in Figure 4.4. At each stage, the proposed location for $\vec{x}_l^{(k)}$ is used in an experiment, and the resulting value of $G(\vec{x}_l^{(k)})$ is used to update the GP fit. Countours of the GP mean, conditioned on observed data, are shown on top. The location that produced the greatest thermal signal, $\vec{x}_l^{+}$ at each iteration is highlighted in magenta, and the true value of the corrosion pit center $\vec{x}_c$ is shown in black. The expected improvement function at each iteration is shown below.

The IR images should be retained from each step of the Bayesian optimization iterations. The measurement with the greatest thermal response, $\mathcal{D}^{+}$, resulting from laser spot $\vec{x}_l^{+}$ (not necessarily the final experiment), is then to be used for the final stage of characterization analysis. If it is desired, the characterization procedure could stop here, having an accurate approximation of where damage is located, but not of its severity. Alternatively, more iterations can be performed if very high confidence is desired. On the other hand, the

initial experiment may result in no significant deviation from the expected thermal response. If this situation is observed, the iterations could be terminated and the detected anomaly ignored. Paired with a lower likelihood threshold in the detection scan stage, a more cautious procedure is obtained, with a small cost for false positive errors.

4.4.5 Detailed Characterization: Markov Chain Monte Carlo

The final stage in the corrosion characterization procedure is the approximation of the distribution of the corrosion pit parameters $\vec{\theta}$ given the optimal experimental data. In this work, we use the *Markov chain Monte Carlo* (MCMC) algorithm to generate dependent samples from the posterior distribution $p(\vec{\theta}|\mathcal{D}^+, \vec{x}_l^+)$ [25]. In short, each sample, $\vec{\theta}$, requires a simulation of the forward model, Equation (4.1), with the specified parameter values. Then the likelihood that the observed data was the result of $\vec{\theta}$ is computed with Equation (4.4), and Bayes' theorem, Equation (4.3), is used to compare the posterior probabilities of these parameterizations. After taking many samples from the posterior distribution, informative statistics can be computed, such as the posterior mean and variance. Furthermore, a histogram of the samples will illustrate more interesting behaviors of the distributions, such as multimodality, skewness, or complicated joint behavior between parameter values. The capability to automatically characterize the posterior distribution to this degree is the reason MCMC methods are chosen for this work. We note that only the single, optimal data $\mathcal{D}^+$ is used for MCMC inference, and all other scan information is discarded. This is done so that separate simulations with proposed corrosion parameters and many different laser spot locations are not necessary, as

the optimal laser spot location has been found that gives the most useful thermal information.

We will next cover the MCMC implementation in more detail. The prior distribution, $p(\vec{\theta})$, is set with some knowledge from the experiment optimization stage, but is generally uninformative. Thus the likelihood distribution from Equation (4.4) can "speak for itself," having already been provided high quality thermal data from the previous stage. The prior distributions for x_c and y_c are set to be uniform over the specimen area for which the final Gaussian process mean fit to $G(\vec{x}_l)$ is positive. The distributions for the corrosion pit penetration depth, d, and width, w, are set to be uniform between 0 mm and 10 mm, essentially covering the entire inspection volume. The sequence of samples is initialized at $\vec{\theta}^{(0)} = (x_l^+, y_l^+, d_0, w_0)$, with the corrosion size parameters sampled at random from their prior distributions. Then Metropolis within Gibbs sampling [41] is used to simultaneously perform inference over all four parameters according to the following iterative scheme.

From sample $\vec{\theta}^{(k)}$, a nearby proposal $\vec{\theta}_*$ is constructed by varying a single parameter π at a time, and drawing a candidate at random according to $(\vec{\theta}_*)_\pi \sim \text{Unif}((\vec{\theta}^{(k)})_\pi - A_\pi/2, (\vec{\theta}^{(k)})_\pi + A_\pi/2)$. In other words, $\vec{\theta}_*$ is constructed by sequentially adjusting each of its parameter entries, $(\vec{\theta}_*)_\pi$, where $\pi \in \{1, 2, 3, 4\}$ and either accepting or rejecting the proposal according to the resulting data fit. The proposal distribution widths $\vec{A}$ have been tuned to minimize the autocorrelation in the sequence of samples [43]. The likelihood $p(\mathcal{D}^+|\vec{\theta}_*,\ \vec{x}_l^+)$ is computed using the simulated ideal solution $F(\vec{x}_l^+, \vec{\theta}_*)$ in Equation (4.4). If we find that the new parameter is more likely, given the data, it is accepted and $(\vec{\theta}^{(k+1)})_\pi = (\vec{\theta}_*)_\pi$.

Otherwise, we accept it anyway with probability

$$\frac{p(\vec{\theta}_*|\mathcal{D}^+,\, \vec{x}_l^+)}{p(\vec{\theta}^{(k)}|\mathcal{D}^+,\, \vec{x}_l^+)} = \frac{p(\mathcal{D}^+|\vec{\theta}_*,\, \vec{x}_l^+)p(\vec{\theta}_*)}{p(\mathcal{D}^+|\vec{\theta}^{(k)},\, \vec{x}_l^+)p(\vec{\theta}^{(k)})}.$$

If the proposal is rejected, then we set $(\vec{\theta}^{(k+1)})_\pi = (\vec{\theta}^{(k)})_\pi$ and proceed through the parameters for $\pi \in \{1, 2, 3, 4\}$, and then on to the next sample. The random behavior of the sample chain, occasionally accepting a "worse" sample according to the specified probability, guarantees, in the long run, that the resulting Markov chain converges to the proper stationary distribution [25]. In practice, a short *burn-in* sample set is typically removed from the beginning of the chain, to ensure that the initial state does not artificially influence the reported behavior of the samples. After the burn-in period, no samples are discarded, since fully independent samples are not necessary.

4.5 Results and Discussion

The current results have been produced on AMD FirePro D700 GPU with 2048 streaming processors, 6GB of onboard memory, and up to 32KB of local memory per work group. Some intermediate results have been shown for illustrative purposes in the previous section, using a particular corrosion pit parameterized by $\vec{\theta}_{\text{true}} = (0, 4, 4, 2)$. Summarizing these earlier results, we note that the damage was flagged by the detection procedure in Section 4.4.3 as an anomaly at $(1.37, 2.55)$ with an estimated standard deviation in the likelihood surface of 23.1 mm. These values were input to the Bayesian experiment optimization method in Section 4.4.4, which refined the estimate to $\vec{x}_c \approx \vec{x}_l^+ = (0.94, 4.16)$. It was claimed in that section that a corrosion pit could be accurately found (within 3 mm) in 7 iterations. Further demonstrations for this claim are provided with

Trial	Distance from Scan Line (mm)	Corrosion Size (d, w) (mm)	Detection Time (s)	Detection Error (mm)	BO Error (mm)
1	4	(4, 2)	2.42	(1.37, -1.45)	(0.40, -1.47)
2	8	(4, 1.5)	3.84	(0.09, -3.45)	(0.42, -0.39)
3	12	(4, 1)	5.94	(0.48, -6.61)	(-0.64, 1.73)
4	4	(2, 4)	2.58	(-1.07, -4.91)	(0.17, 0.27)
5	8	(1.5, 4)	3.48	(2.08, -4.15)	(-0.81, 1.18)
6	12	(1, 4)	4.66	(-4.39, -5.99)	(1.97, 0.46)

Table 4.2: Six corrosion scenarios and the error in estimating their location after the detection stage and Bayesian optimization stage.

five additional simulated trials of varying severity, summarized in Table 4.2 and in Figure 4.7.

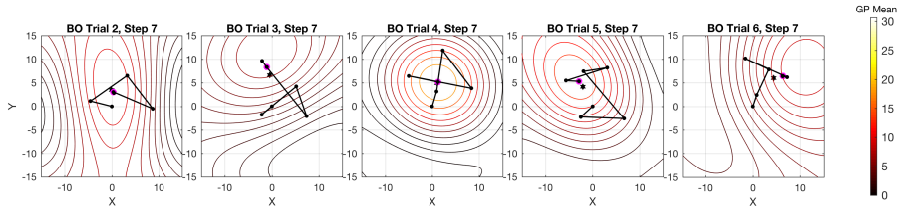

Figure 4.7: Results of Bayesian optimization after six iterations for five additional scenarios with the corrosion pit location and shape set according to Table 4.2. As in Figure 4.6, the black star denotes $\vec{x}_c$, and the laser spot point $\vec{x}_l^+$ which gave the greatest thermal response is highlighted in magenta.

Taking the optimal data from the previously discussed experiments, Markov chain Monte Carlo simulation is performed according to Section 4.4.5. A burn-in period of 1,000 samples is discarded, followed by 20,000 samples that are retained. The lengths of these intervals is observed to be more than sufficient to generate the posterior distribution behavior described below. The estimated marginal posterior distributions for $p(\vec{\theta}|\mathcal{D}^+, \vec{x}_l^+)$ are shown in Figure 4.8, and the posterior statistics summarized in Table 4.3. We see that good estimates for $\vec{x}_c$ have been achieved. Estimates for the corrosion size contain the true values

Corrosion Parameter	True Value	Posterior Mean	Posterior St. Dev
x_c (mm)	0.0	0.3	1.1
y_c (mm)	4.0	3.0	1.2
d (mm)	4.0	3.6	1.6
w (mm)	2.0	2.8	1.4

Table 4.3: Posterior mean and variance MCMC sampling.

Trial	True Parameters (x_c, y_c, d, w)	MCMC Posterior Estimate
1	(0, 4, 4, 2)	$(0.3 \pm 1.1,\ 3.0 \pm 1.2,\ 3.6 \pm 1.6,\ 2.8 \pm 1.4)$
2	(0, 8, 4, 1.5)	$(0.3 \pm 1.4,\ 7.7 \pm 1.6,\ 1.6 \pm 1.1,\ 4.7 \pm 2.2)$
3	(0, 12, 4, 1)	$(-0.2 \pm 2.3,\ 14.4 \pm 2.7,\ 3.1 \pm 2.2,\ 2.3 \pm 2.1)$
4	(0, 4, 2, 4)	$(0.1 \pm 1.1,\ 4.2 \pm 1.2,\ 4.4 \pm 1.3,\ 2.2 \pm 0.7)$
5	(0, 8, 1.5, 4)	$(0.2 \pm 1.2,\ 7.9 \pm 1.3,\ 2.2 \pm 1.3,\ 4.1 \pm 2.1)$
6	(0, 12, 1, 4)	$(1.5 \pm 2.1,\ 12.4 \pm 1.9,\ 5.0 \pm 2.1,\ 1.4 \pm 0.8)$

Table 4.4: Posterior statistics from MCMC sampling.

within one standard deviation, skewed towards a shorter and wider pit. The same analysis is performed for the remaining five scenarios from Table 4.2, with posterior statistics summarized in Table 4.4. The error from progressive estimates of the corrosion pit location throughout the procedure are give in Table 4.6.

The joint distributions of the parameters are illuminating in this process, as shown in Figure 4.9. The location parameters seem uncorrelated, meaning that

Trial	True Value (mm^2)	MCMC Posterior Estimate
1	20	19.6 ± 1.8
2	15.0	13.5 ± 1.4
3	10.0	9.6 ± 1.8
4	20.0	22.3 ± 1.8
5	15.0	16.5 ± 1.5
6	10.0	13.5 ± 1.7

Table 4.5: Posterior mean and variance for cross-sectional area

Trial	Location Error After Detection Scan (mm)	Location Error After BO (mm)	Location Error After MCMC (mm)
1	2.0	1.5	0.8
2	3.4	0.6	0.5
3	6.6	1.8	2.9
4	5.0	0.3	0.2
5	4.6	1.4	0.2
6	7.4	2.0	1.1

Table 4.6: Corrosion pit location error after each stage of analysis.

the vertical and horizontal values do not exhibit dependence, as one would expect. However, there is a high level of dependence in the two size parameters, which is present in all six trials. The relationship shows that the thermographic corrosion characterization procedure which is used to get these distributions is much more sensitive to the *severity* of corrosion, as measured by total corrupted volume, than it is to either its depth or width independently. Such a measurement may also be informative to an inspector who is more interested in the extent of damage than its particular shape. Nevertheless, we have shown that, through an optimized experimental design, thermographic data can be gathered which provides enough information for successful inference over the location and shape of a small hidden region of corrosion. A more efficient sampling procedure could explicitly use corrosion volume and skewness as inference parameters to improve exploration of the posterior distribution, compared to crescent form when corrosion depth and width are used.

4.6 Conclusions

In this paper, we set out a procedure for detecting and characterizing instances of pitting corrosion on inaccessible regions of steel structures. The procedure

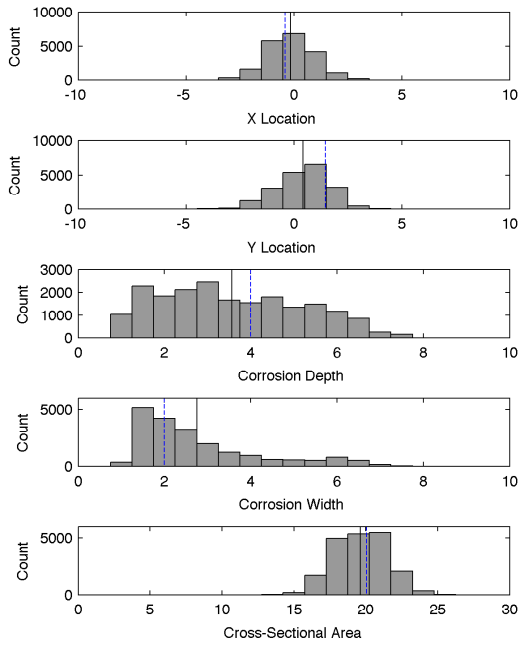

Figure 4.8: Trial 1 histograms of MCMC samples estimating the marginal posterior distributions of each corrosion pit parameter. The true values for each parameter are denoted by a dashed blue line, while the posterior mean is shown in black.

relies on a sequence of thermographic experiments using a laser and an IR camera. Mathematically, a Bayesian inference pipeline is developed and exploited from one stage of the experiments to the next. Prior beliefs over the location and shape of the corrosion pit are updated in a principled way, culminating with a targeted experiment that is designed to be optimal for the purpose of Bayesian inference. The numerical experiments and demonstration of the Bayesian inference methods have been made possible by a fast solver for the heat equation PDE over a heterogeneous material domain.

We conclude with a few remarks on the generalization and practical departures from our proposed inference methods. Although the specific problem of

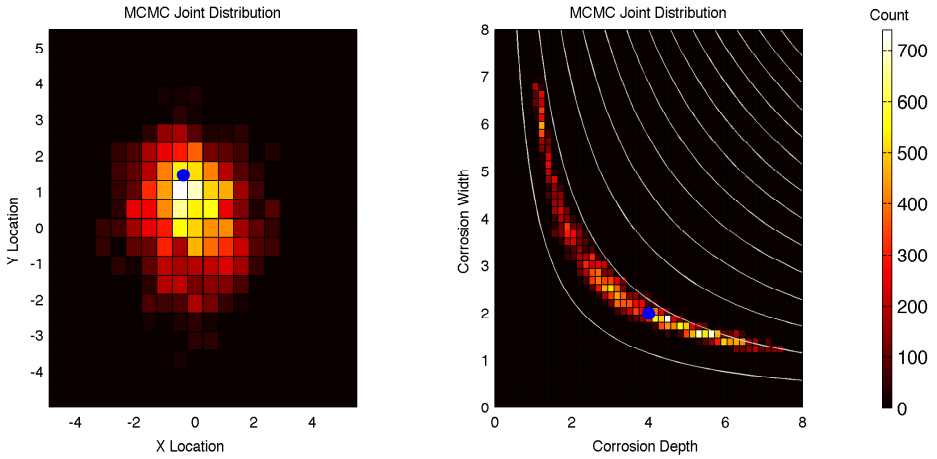

Figure 4.9: Trial 1 joint posterior distributions for two pairs of corrosion pit parameters. (left) The location parameters do not exhibit strong correlation, and (right) the shape parameters do. Contour lines for the total corrosion volume function are overlaid in white. For both plots, the true values from the experimental corrosion pit are shown with blue dots.

finding corrosion pits in steel structures is considered here, the framework that was described is applicable to any passive metallic material. Additionally, the Bayesian inference pipeline can be truncated at any point that a suitable characterization of damage has been found. That is, it may be sufficient to perform only the detection phase before maintenance action is taken; or only one or two iterations of Bayesian experiment optimization may be enough to provide useful information to an inspector. The hypothetical 100 W laser was chosen as an energy source for this work due to its flexibility for wide detection scans, as well as detailed pulses on a fixed location. When investigating a larger structure, we note that the scans and pulses can be done on far away regions, while a recently probed area cools down to the ambient temperature. An optimal laser scheduling algorithm for an entire bridge, for instance, could lead to an efficient large-scale NDT procedure.

APPENDIX A

CHAPTER 1 OF APPENDIX

A.1 PDE Solution

The heat equation PDE with Gaussian laser pulse heating has been previously studied in the literature. Lax presented the time-independent case of the temperature rise induced by a laser beam in a light-emitting semiconductor in 1977 [38]. Li et al. gave a similar time-dependent derivation for the case of an elliptical Gaussian beam heating a 3D structure [39]. The derivation below will be independent of these earlier references, as discussion is limited to the specific case of a circular Gaussian beam heating an infinite 2D plane.

Using the notation from Section 4.3, the non-homogeneous boundary value problem is formulated as

$$\begin{cases} \frac{\partial T(\vec{x},t)}{\partial t} - \kappa \nabla^2 T(\vec{x},t) = f(\vec{x},t)/\rho h C & \text{in } \mathbb{R}^2 \times (0,\infty) \\ T(\vec{x},t) = T_0 & \text{on } \mathbb{R}^2 \times \{t=0\} \end{cases}, \tag{A.1}$$

with

$$f(\vec{x},t) = \frac{2P}{\omega^2} \exp\left(-\frac{2\left|\vec{x}-\vec{x}_l\right|^2}{\omega^2}\right).$$

The Green's function for this PDE is

$$\Phi(\vec{x},t) = \frac{1}{4\pi t\kappa} \exp\left(-\frac{|\vec{x}|^2}{4t\kappa}\right)$$

satisfying

$$\begin{cases} \frac{\partial \Phi(\vec{x},t)}{\partial t} - \kappa \nabla^2 \Phi = 0 & \text{in } \mathbb{R}^2 \times (0,\infty) \\ \Phi(\vec{x},0) = \delta_0 & \text{on } \mathbb{R}^2 \times \{t=0\}. \end{cases}$$

Then convolution of the forcing function and the Green's function $\hat{T}(\vec{x},t;s) = \Phi(\vec{x},t-s) * f(\vec{x},s)$ satisfies the parameterized initial value problem

$$\begin{cases} \frac{\partial \hat{T}(\vec{x},t;s)}{\partial t} - \kappa \nabla^2 \hat{T} = 0 & \text{in } \mathbb{R}^2 \times (0,\infty) \\ \hat{T}(\vec{x},t;s) = f(\vec{x},s) & \text{on } \mathbb{R}^2 \times \{t=0\}. \end{cases}$$

Duhamel's Principle states that the forcing function can be viewed as a continuous application of impulses [19], so that $T(\vec{x},t) = \int_0^t \hat{T}(\vec{x},t;s)ds + T_0$ satisfies the given PDE. Thus

$$\begin{aligned} T(\vec{x},t) &= \int_0^t \hat{T}(\vec{x},t;s)ds + T_0 \\ &= \frac{2P}{\pi k}\int_0^t \frac{\exp\left(-\frac{2|\vec{x}-\vec{x}_l|^2}{\omega^2+8\kappa(t-s)}\right)}{\frac{\omega^2+8\kappa(t-s)}{\kappa}}ds + T_0 \\ &= \frac{P}{4\pi k}\int_{\frac{2r^2}{8\kappa t+\omega^2}}^{\frac{2r^2}{\omega^2}} \frac{\exp(-s)}{s}ds + T_0 \\ &= \frac{P}{4\pi k}\left(\text{Ei}\left(-\frac{2r^2}{\omega^2}\right) - \text{Ei}\left(-\frac{2r^2}{\omega^2+8\kappa t}\right)\right) + T_0, \end{aligned}$$

where $\text{Ei}(z) = -\int_{-z}^{\infty} \frac{\exp(-t)}{t}\,dt$ and $r = |\vec{x}-\vec{x}_l|$.

To visualize this function, we use a dimensional reduction method to form a scaled temperature response [27]. Within this, we aggregate all of the material and laser parameters into a single distance-like, and a single time-like variable, so that the general temperature response can be plotted in three dimensions. We first find the proper temperature scaling factor. Let $T_c(t) = \lim_{r\to 0} T(r,t)$. Then

$$\begin{aligned} T_c &= \lim_{r\to 0}\left[\frac{P}{4\pi k}\left(\text{Ei}\left(-\frac{2r^2}{\omega^2}\right) - \text{Ei}\left(-\frac{2r^2}{\omega^2+8\kappa t}\right)\right) + T_0\right] \\ &= \frac{P}{4\pi k}\ln\left(\frac{8\kappa t}{\omega^2}+1\right) + T_0 \\ &= A(t)\frac{P}{4\pi k} + T_0, \end{aligned}$$

and $\overline{T} = (T-T_0)/T_c$ is the scaled temperature function. Note also that this expression for T_c affords a way to compute the peak temperature of the response,

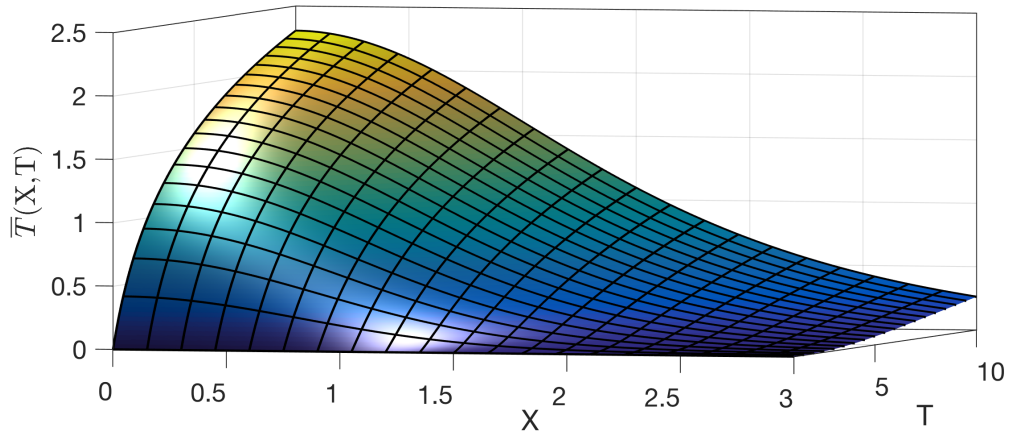

Figure A.1: Surface plot of the nondimensionalized response as a function of $\mathcal{X}$ and $\mathcal{T}$.

without evaluating an integral. Next, we define a dimensionless variable to represent the distance of a point to the laser peak position, $\mathcal{X} = 2r/\omega$, which scales linearly with r. Finally, we define $\mathcal{T} = (\omega^2 + 8\kappa t)/\omega^2$, which scales linearly with t, inspired by the expression of $T(r,t)$. The nondimensionalized temperature response is then

$$\overline{T}(\mathcal{X},\mathcal{T}) = A\int_{\mathcal{X}^2/\mathcal{T}}^{\mathcal{X}^2} \frac{e^{-z}}{z}dz.$$

Figure A.1 shows a surface plot of this function for $A = 1$. The temperature response for any particular system is found by calculating A, $\mathcal{X}$, and $\mathcal{T}$ from the desired parameters and sweeping $\overline{T}(\cdot,\mathcal{T})$ over a circle to make a radially symmetric response.

A.2 MCMC Details

The Markov chain Monte Carlo method is developed for the efficient application of Bayes' rule for conditional probability. Define $\vec{\theta}$ to be a vector of val-

ues for the parameters of interest (e.g. model parameters instantiating a given flaw context), $\vec{T}$ to be the actual temperature response at all measured spatio-temporal locations, and $\vec{T}^* = \vec{T} + \vec{\eta}$ to be the measured response under $\vec{\eta}$ assumed measurement noise. We seek the *probability density function* of the crack parameters given a particular noisy thermal measurement, $p(\vec{\theta}|\vec{T}^*)$, the *posterior distribution*. Bayes' rule states that this is proportional to the product of $p(\vec{T}^*|\vec{\theta})$ (the *likelihood* distribution of a measured response given a particular set of crack parameters) and $p(\vec{\theta})$ (the *prior* distribution which contains all the beliefs regarding plausible crack parameters).

In our case, we make minimal assumptions regarding the crack elliptical geometry, and the prior distribution is uniform over a wide range of values for each subsequent parameter (i.e. their support). Our measured response is taken over a 3×3 pixel patch, within which the crack is assumed to be located; thus the prior distributions for the horizontal and vertical location of the crack center are uniform over this area, while the prior distribution of the crack length is taken to be uniform from zero to the length of one pixel. We also assume that the additive noise $\vec{\eta}$ over each pixel is i.i.d. Gaussian, with a standard deviation equal to the NETD of the considered imaging system, $p(\eta_i) = (2\pi\sigma)^{-1/2} \exp(-\eta_i^2/(2\sigma))$. This PDF is to be used in forming the likelihood distribution.

We wish to be able to evaluate the likelihood function $p(\vec{T}^*|\vec{\theta})$ for many values of $\vec{\theta}$. To do this, we use the finite element modeling software to simulate a temperature response over a specified crack instance, $\vec{T}(\vec{\theta})$ and substitute this response into the likelihood function (i.e. the likelihood is closed within a computer model). Then the probability that our noisy thermal measurement was caused by this crack is governed by the probability that each pixel reading had

realized the difference between its measured value and the simulated response as measurement noise

$$\begin{aligned} p(\vec{T}^*|\vec{\theta}) &= p_\eta \left(T_1(\vec{\theta}) - T_1^*, \ldots, T_9(\vec{\theta}) - T_9^* \right) \\ &= \prod_i p_\eta \left(T_i(\vec{\theta}) - T_i^* \right) \\ &= \prod_i \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(T_i(\vec{\theta}) - T_i^*)^2}{2\sigma} \right) \\ &= \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\sum_i (T_i(\vec{\theta}) - T_i^*)^2}{2\sigma} \right). \end{aligned}$$

We have now defined the necessary components to perform MCMC sampling of the posterior distribution. At each step k in the chain, we have evaluated the likelihood $p^{(k)} = p(\vec{T}^*|\vec{\theta}^{(k)})$. A candidate $\vec{\theta}^*$ is chosen randomly according to a multivariate uniform distribution with width $2\vec{L}$ centered at $\vec{\theta}^{(k)}$, and the candidate likelihood $p^* = p(\vec{T}^*|\vec{\theta}^*)$ is computed. The ratio of these two likelihoods determines how the chain moves. If the new location $\vec{\theta}^*$ has higher likelihood than the previous location, then it is selected as $\vec{\theta}^{(k+1)}$. Otherwise, the chain moves to the new location with probability $p^*/p^{(k)}$ and remains put, $\vec{\theta}^{(k+1)} = \vec{\theta}^{(k)}$ with probability $1 - p^*/p^{(k)}$. This permission to move to a less likely parameter is the key to the Metropolis-Hastings algorithm. Under our condition that the chain can transition from any state $\vec{\theta}$ to any other state $\vec{\theta}'$ in a finite number of steps (*irreducibility* of the chain), the MCMC samples are guaranteed to converge to a unique, stationary posterior distribution [25].

In our inverse problem solution, we simultaneously solve for three crack parameters encoded in $\vec{\theta}$. In practice, each parameter is updated one-at-a-time, so every step of the chain requires three evaluations of $p^{(k)}$, with one updated parameter each time. This process is known as *Metropolis-within-Gibbs sampling*. The estimated PDF of each crack parameter is simply the marginal distribution

of the MCMC samples.

The convergence of MCMC samples occurs at a faster rate if subsequent samples have low correlation. Typically, a *burn-in* period is used to ensure independence of the sample from the initial location. We also use the burn-in period to tune the width of the candidate distribution for each parameter, L [41]. A narrow candidate distribution will result in relatively high probability of accepting the new candidate at each step, but the candidates will be close to the previous samples. On the other hand, a wide candidate distribution can provide distant candidates, but a lower acceptance probability implies that many subsequent samples will be identical. Balancing these factors motivates the tuning of L for each chain during burn-in. By increasing L by a small factor after each accepted burn-in sample and decreasing it by some related factor after each rejection, it can be tuned to find a value associated with a certain probability of acceptance. The literature suggests a target acceptance probability of 0.42 [41], but this was found to produce samples with very high correlation in our three parameter chains. We performed several test chains with fixed values of L for the uncracked problem to determine the proper acceptance probability for our problem. Figure A.2 shows empirical lag 1 autocorrelations and acceptance probabilities for varying values of L. It is determined that an acceptance probability near 0.1 is associated with minimal autocorrelation for all three crack parameters; thus this value was used as the target for the burn-in periods of all MCMC simulations presented in this work.

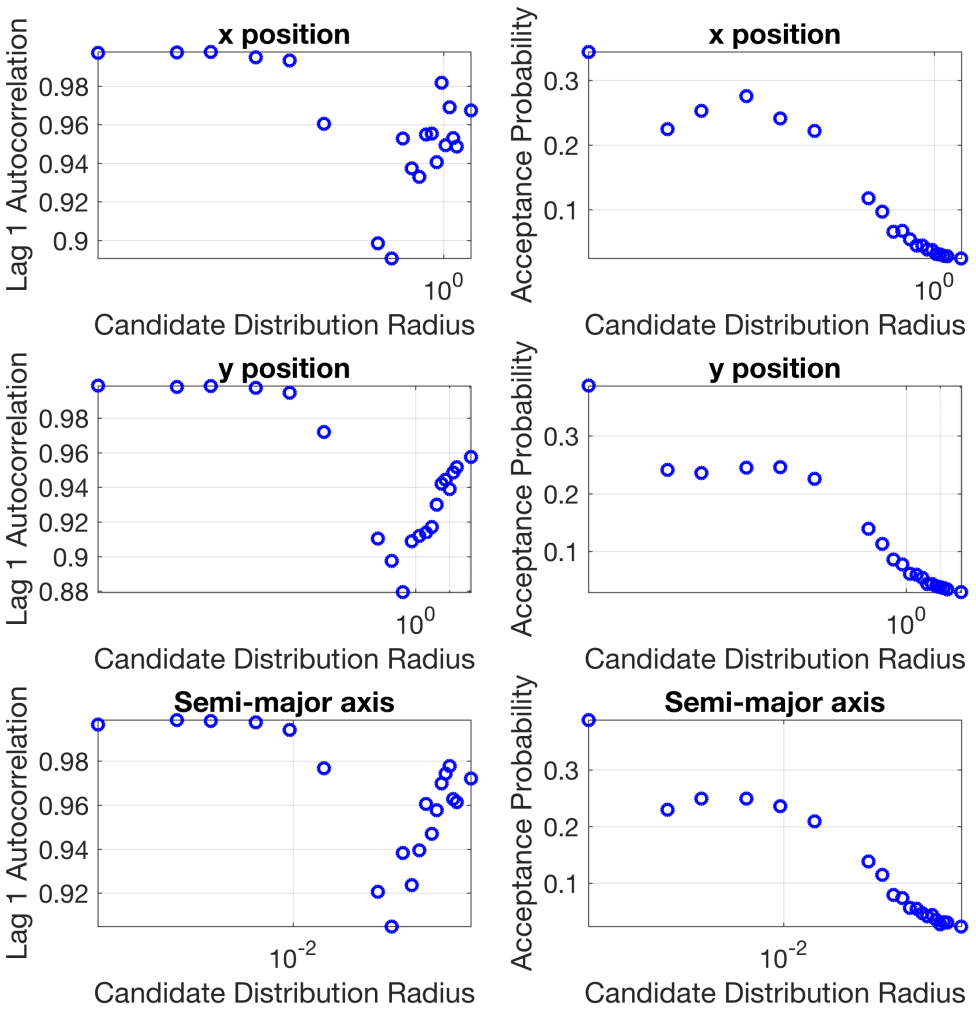

Figure A.2: (left) Experimental lag 1 autocorrelation as a function of L for the for uncracked problem. (right) Experimental acceptance probability of the same data.

A.3 Convergence Analysis

In order to be confident that our MCMC samplings are sufficiently long to provide good estimates of the posterior distribution $p(\vec{\theta}|\vec{T}^*)$, we carry out a separate, independent convergence study. We analyze here two sets of MCMC samplings, each having a single noise realization across $m = 10$ trials. The m chains are initialized randomly across the support of the prior distribution, with each sampled uniformly from a disjoint interval spanning one m^{th} of the width of the prior. This is so that we have confidence that the MCMC samples that we collect after the burn-in period are valid from any random initial point. Each chain has 5,000 discarded burn-in samples and 10,000 saved samples.

The convergence diagnostic we use is based on $(1 - \alpha)$ credible intervals [6]. From each individual chain, the empirical $(1 - \alpha)$ credible intervals are computed. For each within-chain interval, the proportion of samples from all m chains that occur within this interval is determined. Finally, these proportions are averaged to give the convergence diagnostic. If all m chains have converged to the same distribution, then the convergence diagnostic will converge to $(1 - \alpha)$. Figure A.3 shows trajectories of this value as the first set of simulations evolves with $\alpha = 0.05$. The convergence diagnostic is seen to converge to $(1 - \alpha)$ within the 10,000 samples used. The behavior is the same with the second set of simulations and for any value of α. We are therefore confident that the simulations presented in Section 2.4.4, which have 20,000 samples, are reliable estimates of the desired distribution.

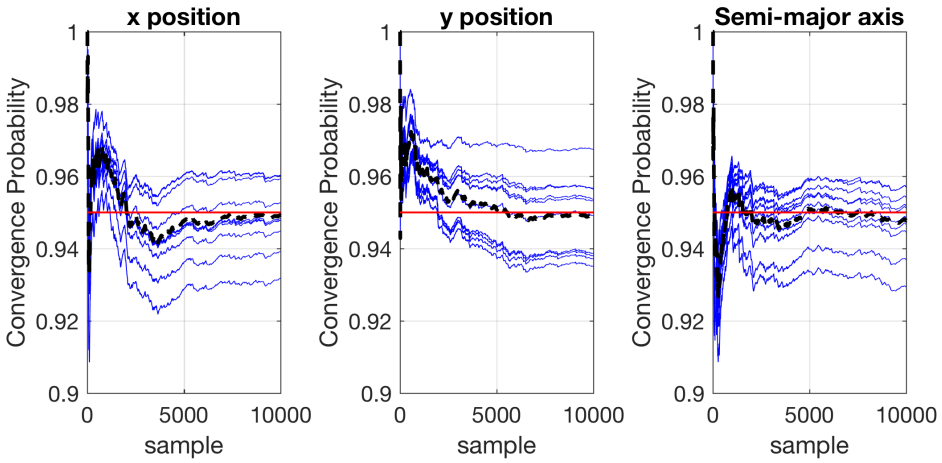

Figure A.3: Trajectories of the $(1-\alpha)$ convergence diagnostic for a representative pool of MCMC samplings.

APPENDIX B

CHAPTER 2 OF APPENDIX

B.1 GPU PCG Algorithm Details

This appendix contains descriptions of the custom kernels that were written to perform linear algebra operations, how they use GPU memory, and how they fit into the PCG algorithm.

B.1.1 Memory

We first introduce the FEM data and indexing variables that are used throughout our kernels:

- M and F: arrays of data for elemental assembly matrices. Either the single matrices for fixed grid methods, or all entries for every element for the general methods
- DoFMapLocal and coordinateMapLocal: small arrays of indices based on the geometry of the mesh to quickly determine local DoF to element assignment
- C: array of the number of divisions of the domain in each dimension
- vert_scale: array of the minimum vertex position and spacing between vertices in each dimension. Along with C, this allows the determination of the absolute location of a vertex given its global index. For a non-fixed grid approach, the user may start with a uniform mesh and deform it in

a predetermined way (e.g. quadratically scaling the vertex locations), so that information can be used inside a kernel to still enable the recovery of absolute vertex position.

- corr_bound: spatial information about what part of the domain is corroded (or in general, a different material). For uniform corrosion after a certain depth, this can be that distance in the z-direction. For elliptical corrosion pits, it could contain the coordinates of the center of the ellipse and its axis lengths. The kernel must be programmed to know how to interpret this.
- mat_coefs: values for ρC and k for the different materials

Memory is allocated in GPU global memory buffers with flags to specify how the host and the device will access them. The flags are self-explanatory, and are combined with logical OR. The combinations used here are:

- HOST_TO_DEVICE_COPY = (READ_ONLY | HOST_WRITE_ONLY | COPY_HOST_PTR)
- HOST_TO_DEVICE_USE = (READ_ONLY | HOST_WRITE_ONLY | USE_HOST_PTR)
- HOST_READ_WRITE = (READ_WRITE | COPY_HOST_PTR)
- PINNED = (READ_WRITE | USE_HOST_PTR)
- DEVICE_READ_WRITE = (READ_WRITE | HOST_NO_ACCESS)

The memory buffers allocated for the FG DbD method are enumerated in Table B.1. Other methods do not differ much at this level.

Name	Flag	Initialization
M_FG_buf	HOST_TO_DEVICE_USE	M_FG
K_FG_buf	HOST_TO_DEVICE_USE	K_FG
DoFMapLocal_buf	HOST_TO_DEVICE_COPY	DoFMapLocal
coordinateMapLocal_buf	HOST_TO_DEVICE_COPY	coordinateMapLocal
C_buf	HOST_TO_DEVICE_COPY	C
vert_scale_buf	HOST_TO_DEVICE_COPY	vert_scale
corr_bounds_buf	HOST_TO_DEVICE_COPY	corr_bounds
mat_coefs_buf	HOST_TO_DEVICE_COPY	mat_coefs
P_buf	DEVICE_READ_WRITE	x.nbytes
u0_buf	HOST_TO_DEVICE_COPY	u0
Fdt_buf	HOST_TO_DEVICE_COPY	Fdt
x_buf	HOST_READ_WRITE	x
b_buf	DEVICE_READ_WRITE	x.nbytes
r_buf	DEVICE_READ_WRITE	x.nbytes
d_buf	DEVICE_READ_WRITE	x.nbytes
q_buf	DEVICE_READ_WRITE	x.nbytes
s_buf	DEVICE_READ_WRITE	x.nbytes
delta_buf	PINNED	delta
alpha_buf	DEVICE_READ_WRITE	delta.nbytes
neg_alpha_buf	DEVICE_READ_WRITE	delta.nbytes
delta_new_buf	DEVICE_READ_WRITE	delta.nbytes
beta_buf	DEVICE_READ_WRITE	delta.nbytes
r1_buf	DEVICE_READ_WRITE	r1_size×4
r2_buf	DEVICE_READ_WRITE	r2_size×4
VVM_loc_buf	LocalMemory	max_wg_size×4
Ax_split_buf	DEVICE_READ_WRITE	x.nbytes×4
P_split_buf	DEVICE_READ_WRITE	x.nbytes×4
x_local_buf	LocalMemory	32×4×4

Table B.1: Memory buffer types and initialization values. If a number is given for initialization, the specified number of bytes is allocated.

B.1.2 Kernels

VVM_A

First stage of a vector-vector multiplication. Takes pairwise scalar products of vector elements and then uses a standard parallel "reduction" algorithm to sum

the results in $\mathcal{O}(\log(N))$ complexity. The work group size is maximized for the available hardware, and the partial sum is reduced by a factor of two at each step. The total number of elements in the partial sum can only be reduced by a factor of the maximum work group size with a single kernel call. In case the length of the vector, nVertices, is not a multiple of the maximum work group size, max_wg_size, the number of work groups is rounded up. The size of the result (number of elements) is then

$$\text{r1\_size} = \left\lceil \frac{\text{nVertices}}{\text{max\_workgroup\_size}} \right\rceil .$$

Argument	**In/Out**	**Memory space**	**Data type**	**Size**
*x	input	global	float64	nVertices
*y	input	global	float64	nVertices
nVertices	input		uint32	1
*VVM_loc		local	float64	max_workgroup_size
*r	output	global	float64	r1_size
global size	r1_size × max_wg_size	**local size**	max_wg_size	

VVM_reduce

Intermediate and final stages of vector-vector multiplication. Takes an intermediate partial sum from VVM_A or itself and reduces it by a factor of max_workgroup_size. If nVertices $\leq$ max_workgroup_size, the vector-vector multiplication is completed. Otherwise further calls with this kernel are made. Since this can be iterative, set

$$\text{rk\_size} = \left\lceil \frac{\text{r(k-1)\_size}}{\text{max\_workgroup\_size}} \right\rceil .$$

Argument	In/Out	Memory space	Data type	Size
*rPrevious	input	global	float64	r(k-1)_size × max_workgroup_size
nVertices	input		uint32	1
*VVM_loc		local	float64	max_workgroup_size
*r	output	global	float64	rk_size
global size	rk_size × max_wg_size	**local size**	max_wg_size	

VVM_C

An alternative final stage for vector-vector multiplication for step 5 of PCG. Rather than storing the final result of the sum, the scalar delta_new is loaded and alpha = delta_new/($d^T q$) is stored, along with negative alpha.

Argument	In/Out	Memory space	Data type	Size
*rPrevious	input	global	float64	r(k-1)_size × max_workgroup_size
*delta	input	global	float64	1
nVertices	input		uint32	1
*b		local	float64	max_workgroup_size
*alpha	output	global	float64	1
*neg_alpha	output	global	float64	1
global size	rk_size	**local size**	max_wg_size	

VAVSP

Computes the elementwise sum of a vector and a scalar multiplied by another vector. The scalars are loaded from GPU global memory, so a negative scalar must be used if vector subtraction is desired.

Argument	In/Out	Memory space	Data type	Size
*x	input	global	float64	nVertices
*y	input	global	float64	nVertices
*a	input	global	float64	1
*x_plus_ay	output	global	float64	nVertices
global size	nVertices	**local size**	None	

DIMVM

Computes the matrix-vector multiplication where the matrix is the inverse of a diagonal matrix P. Element i of the result is x_i/P_{i,i}.

Argument	In/Out	Memory space	Data type	Size
*P	input	global	float64	nVertices
*x	input	global	float64	nVertices
*Pinvx	output	global	float64	nVertices
global size	nVertices	**local size**	None	

Beta_update

Computes the coefficient beta and updates delta in storage. Used to avoid data transfer between device and host for the performance of a small calculation.

Argument	In/Out	Memory space	Data type	Size
*delta_new	input	global	float64	1
*delta_old	both	global	float64	1
*beta	output	global	float64	1
global size	1	**local size**	1	

u0_update

Computes an initial guess for the next time step, u_0^+, based on a linear extrapolation from the initial guess of the current time step, u_0 and the PCG solution of the current time step, u_new,

$$u\_0^+ = u\_new + (u\_new - u\_0).$$

Argument	In/Out	Memory space	Data type	Size
*u0	both	global	float64	nVertices
*u_new	input	global	float64	nVertices
global size	nVertices	**local size**	None	

FGDbDMVM_A

The implementation of matrix-vector multiplication (including cases for $\mathbf{A}\vec{x}$ and $a\mathbf{A}\vec{x} + \vec{b}$) and the determination of P is dependent on the assembly perspective. The kernels for the third implementation (FG DbD with memory coalescing) are discussed here.

The kernel computes contributions to a matrix-vector multiplication from each element according to coalesced memory access limitations. Data is loaded from the input vector as described in Section 3.4.3 and stored in local memory. Then each work item determines its global element index by first finding the cube in which it belongs: global_id/6 - group_id (global_id is an unsigned integer, so integer division automatically rounds down), and then the tetrahedron within the cube: local_id mod 6.

The cube index also corresponds to the global vertex index for its first corner. This is used with the arrays C and vert_scale to determine the vertex's absolute position. Based on this, the array coordinateMapLocal helps give the x, y, and z positions of the other vertices of the tetrahedron, so that they can be compared with corr_bounds to determine for each vertex whether it is in the corroded region of the domain or not. Material coefficients are taken from mat_coefs and averaged over the element for both $\mathbf{M}_e$ and $\mathbf{K}_e$.

Next, each element reads the data it needs about the input vector from local memory, referring to DoFMapLocal for the proper indices. This has been de-

layed as long as possible to hide the latency from the global memory load. Dot products are taken with the local assembly matrices, and the results are summed with the proper coefficients in another local memory array, Ax_split_local, which has 12 entries for every DoF, corresponding to the 12 possible contributing elments in the $\pm x$ directions. Finally, the work items that are responsible for loading and storing data sum over Ax_split_local for their DoF and write to global memory. The resulting memory buffer has 4 entries for every global DoF, one for every quadrant in the y-z plane. These are all filled out in turn by further work groups.

The total number of work items needed for this kernel is found by determining the total number of elements that need to be considered including the padding in $+x$ and $+y$ directions, dividing by 30 since every work group yields the elemental contributions for blocks of 30 vertices, and multiplying by the work group size. As with vector-vector multiplications, we round up the integer division

$$\text{MVM\_global\_size} = 186 \left\lceil \frac{6(C[0]+1)(C[1]+1)C[2]}{180} \right\rceil .$$

FGDbDMVM_B

Finishes the matrix-vector multiplication started by FGDbDMVM_A. Work items sum the four contributions to each DoF from Ax_split and store them in the final result array.

Argument	In/Out	Memory space	Data type	Size
*M_FG	input	constant	float64	4
*K_FG	input	constant	float64	4
*x	input	global	float64	nVertices
*DoFMapLocal	input	constnat	uint32	12
*coordinateMapLocal	input	constant	float64	12
*C	input	constant	uint32	3
*vert_scale	input	constant	float64	6
*corr_bounds	input	constant	float64	
*mat_coefs	input	constant	float64	4
theta	input		float64	1
dt	input		float64	1
*x_local		local	float64	4×32
*Ax_split_local		local	float64	4×32×12
*Ax_split	output	global	float64	nVertices×4
global size	MVM_global_size	**local size**	186	

Argument	In/Out	Memory space	Data type	Size
*Ax_split	input	global	float64	nVertices×4
*Ax	output	global	float64	nVertices
global size	MVM_global_size	**local size**	186	

FGDbDMVM_C

Finishes the matrix-vector multiplication started by FGDbDMVM_A with an extra SAXPY operation so that a separate call to VAVSM is not necessary. Work items sum the four contributions to each DoF from Ax_split, multiply them by a scalar, add them to an element from another vector, and store the result.

Argument	In/Out	Memory space	Data type	Size
*Ax_split	input	global	float64	nVertices×4
*b	input	global	float64	nVertices
c	input		float64	1
*cAx_plus_b	output	global	float64	nVertices
global size	MVM_global_size	**local size**	186	

Jacobi_A

Determines contributions to the Jacobi preconditioner P. The algorithm is the same as FGDbDMVM_A, except instead of taking dot products with elemental assembly matrices and an input vector, the diagonal elements of the elemental assembly matices are scaled according to material coefficients, combined, and stored. Calling FGDbDMVM_B on the result produces the diagonal of P.
Arguments for this kernel are the same as for FGDbDMVM_A, except without the need for *x and *x_local.

B.1.3 PCG Again

The kernels and memory usage is as follows, folowing the steps in Section 3.3.3. The syntax below is a small abbreviation of the actual PyOpenCL code, and has the form
kernel_instance = kernel_name(args)

First compute P and the right hand side vector $\vec{b} = \mathbf{L}\vec{u}_i + \vec{F}$.

- knl_PA = Jacobi_A(M_FG_buf, K_FG_buf, DoFMapLocal_buf, coordinateMapLocal_buf, C_buf, vert_scale_buf, corr_bounds_buf, mat_coefs_buf, theta, dt, Ax_split_local_buf, P_split_buf)
- knl_PB = FGDbDMVM_B(P_split_buf, P_buf)
- knl_RHS_A = FGDbDMVM_A(M_FG_buf, K_FG_buf, u0_buf, DoFMapLocal_buf, coordinateMapLocal_buf, C_buf, vert_scale_buf, corr_bounds_buf, mat_coefs_buf, (1-theta), dt, x_local_buf, Ax_split_local_buf, Ax_split_buf)

- knl_RHS_B = FGDbDMVM_C(Ax_split_buf, Fdt_buf, np.float64(1), b_buf)

Setup for PCG

- knl_1A = FGDbDMVM_A(M_FG_buf, K_FG_buf, x_buf, DoFMapLocal_buf, coordinateMapLocal_buf, C_buf, vert_scale_buf, corr_bounds_buf, mat_coefs_buf, theta, dt, x_local_buf, Ax_split_local_buf, Ax_split_buf)
- knl_1B = FGDbDMVM_C(Ax_split_buf, b_buf, np.float64(-1), r_buf)
- knl_2 = DIMVM(P_buf, r_buf, d_buf)
- knl_3A = VVM_A(r_buf, d_buf, nVertices, VVM_loc_buf, r1_buf)
- knl_3B = VVM_reduce(r1_buf, r1_size, VVM_loc_buf, r2_buf)
- knl_3C = VVM_reduce(r2_buf, r2_size, VVM_loc_buf, delta_buf)

Perform one PCG iteration

- knl_4A = FGDbDMVM_A(M_FG_buf, K_FG_buf, d_buf, DoFMapLocal_buf, coordinateMapLocal_buf, C_buf, vert_scale_buf, corr_bounds_buf, mat_coefs_buf, theta, dt, x_local_buf, Ax_split_local_buf, Ax_split_buf)
- knl_4B = FGDbDMVM_B(Ax_split_buf, q_buf)
- knl_5A = VVM_A(d_buf, q_buf, nVertices, VVM_loc_buf, r1_buf)
- knl_5B = VVM_reduce(r1_buf, r1_size, VVM_loc_buf, r2_buf)
- knl_5C = VVM_C(r2_buf, delta_buf, r2_size, VVM_loc_buf, alpha_buf, neg_alpha_buf)
- knl_6 = VAVSM(x_buf, d_buf, alpha_buf, x_buf)

- knl_7A = FGDbDMVM_A(M_FG_buf, K_FG_buf, x_buf, DoFMapLocal_buf, coordinateMapLocal_buf, C_buf, vert_scale_buf, corr_bounds_buf, mat_coefs_buf, theta, dt, x_local_buf, Ax_split_local_buf, Ax_split_buf)
- knl_7B = FGDbDMVM_C(Ax_split_buf, b_buf, np.float64(-1), r_buf)
- knl_7 = VAVSM(r_buf, q_buf, neg_alpha_buf, r_buf)
- knl_8 = DIMVM(P_buf, r_buf, s_buf)
- knl_9A = VVM_A(r_buf, s_buf, nVertices, VVM_loc_buf, r1_buf)
- knl_9B = VVM_reduce(r1_buf, r1_size, VVM_loc_buf, r2_buf)
- knl_9C = VVM_reduce(r2_buf, r2_size, VVM_loc_buf, delta_new_buf)
- knl_10 = Beta_update(delta_new_buf, delta_buf, beta_buf)
- knl_11 = VAVSM(s_buf, d_buf, beta_buf, d_buf)
- knl_u0 = u0_update(u0_buf, x_buf)

The CPU handles logical decisions and calls these kernels according to the algorithm until convergence is realized.

BIBLIOGRAPHY

[1] Highway accident report: Collapse of I-35w highway bridge. Technical report, National Transportation Safety Board, 2008.

[2] Understanding infrared camera thermal image quality. Technical report, Electrophysics Corporation, 2011.

[3] J. Alda. Laser and gaussian beam propagation and transformation. *Encyclopedia of optical engineering*, pages 999–1013, 2002.

[4] R Ash, R M Barrer, and D G Palmer. Diffusion in multiple laminates. *British Journal of Applied Physics*, 16(6):873–884, Jun 1965.

[5] E. Brochu, V. Cora, and N. De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[6] S. P. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, 1998.

[7] K. Bryan. Efficient computational methods for thermal imaging of small cracks in plates. Present at the 2012 SIAM Annual Meeting, 2012.

[8] Kurt Bryan and Jr. Lester F. Caudill. An inverse problem in thermal imaging. *SIAM Journal on Applied Mathematics*, 56(3):715–735, 1996.

[9] S.E. Burrows, S. Dixon, S.G. Pickering, T. Li, and D.P. Almond. Thermographic detection of surface breaking defects using a scanning laser source. *NDT&E International*, 44(7):589–596, 2011.

[10] X. Cai. Overlapping domain decomposition methods. In *Advanced Topics in Computational Partial Differential Equations: Numerical Methods and Diffpack Programming*, pages 57–95. Springer Science and Business Media, 2003.

[11] G. F. Carey, E. Barragy andR. McLay, and M. Sharma. Element-by-element vector and parallel computations. *Numerical Methods in Biomedical Engineering*, 4(3):299–307, 1988.

[12] M. Charnley and A. Rzeznik. Thermal detection of inaccessible corrosion. Article, Rose Hulman Institute of Technology, 2014.

[13] K. Chatterjee, S. Tuli, S. G. Pickering, and D. P. Almond. A comparison of the pulsed, lock-in and frequency modulated thermography nondestructive evaluation techniques. *NDT&E International*, 44(7):655–667, 2011.

[14] NVIDIA Corporation. Opencl best practices guide, 2010.

[15] G. Deolmi, F. Marcuzzi, S. Marinetti, and S. Poles. Numerical algorithms for an inverse problem of corrosion detection. *Communications in Applied and Industrial Mathematics*, 1(2):78–98, 2010.

[16] Peng Du, Rick Weber, Piotr Luszczek, Stanimire Tomov, Gregory Peterson, and Jack Dongarra. From cuda to opencl: Towards a performance-portable solution for multi-platform gpu programming. *Parallel Computing*, 38(8):391–407, 2012.

[17] C.J. Earls. Stochastic inverse thermographic characterization of sub-pixel sized through cracks. *Mechanical Systems and Signal Processing*, 30(1):146–156, 2012.

[18] C.J. Earls. Bayesian inference of hidden corrosion in steel bridge connections: Non-contact and sparse contact approaches. *Mechanical Systems and Signal Processing*, 41(1-2):420–432, Dec 2013.

[19] L. Evans. *Partial Differential Equations*. American Mathematical Society, 2010.

[20] G. S. Frankel. Pitting corrosion of metals. *Journal of The Electrochemical Society*, 145(6):2186–2198, 1998.

[21] A. Friedman. *Partial Differential Equations of Parabolic Type*. Dover Books on Mathematics. Dover Publications, 2008.

[22] Jacob Gardner, Matt Kusner, Kilian Q. Weinberger, John Cunningham, and Zhixiang Xu. Bayesian optimization with inequality constraints. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 937–945. JMLR Workshop and Conference Proceedings, 2014.

[23] G. H. Golub and Q. Ye. Inexact preconditioned conjugate gradient method with inner-outer iteration. *SIAM Journal on Scientific Computing*, 21(4):1305–1320, 1999.

[24] Gene H Golub and Charles F Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.

[25] W.K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1969.

[26] T Hohage, M-L Rapún, and F-J Sayas. Detecting corrosion using thermal measurements. *Inverse Problems*, 23(1):53–72, 2006.

[27] M. H. Holmes. *Introduction to the Foundations of Applied Mathematics*, volume 56 of *Texts in Applied Mathematics*. Springer New York, 2009.

[28] T. Hughes, I. Levit, and J. Winget. An element-by-element solution algorithm for problems of structural and solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 36(2):241–254, 1983.

[29] T. Hughes, I. Levit, and J. Winget. Elementbyelement implicit algorithms for heat conduction. *Journal of Engineering Mechanics*, 109(2):576–585, 1983.

[30] Victor Isakov. *Inverse Problems for Partial Differential Equations*. Springer International Publishing, 2017.

[31] JC Jaeger. Conduction of heat in composite slabs. *Quarterly of Applied Mathematics*, 8(2):187–198, 1950.

[32] W.Y. Jeong, C.J. Earls, W.D. Philpot, and A.T. Zehnder. Inverse thermographic characterization of optically unresolvable through cracks in thin metal plates. *Mechanical Systems and Signal Processing*, 27(1):634–650, 2012.

[33] M. Kaviany. *Principles of Heat Transfer*. John Wiley & Sons, Inc., 2002.

[34] I. Kiss, Z. Badics, S. Gyimóthy, and J. Pávó. High locality and increased intra-node parallelism for solving finite element models on gpus by novel element-by-element implementation. In *2012 IEEE Conference on High Performance Extreme Computing*, pages 1–5. Institute of Electrical & Electronics Engineers (IEEE), 2012.

[35] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov, and A. Fasih. Pycuda and pyopencl: A scripting-based approach to gpu run-time code generation. *Parallel Computing*, 38(3):157–174, 2012.

[36] J.C. Krapez, C. Gruss, R. Huttner, F. Lepoutre, and L. Legrandjacques. La caméra photothermique - partie i : Principe, modélisation, application à la détection de fissures. *Instrumentation, Mesure, Métrologie*, 1(1):9–40, 2001.

[37] J.C. Krapez, F. Lepoutre, R. Huttner, C. Gruss, L. Legrandjacques, M. Piriou, J. Gros, D. Gente, S. Hermosilla-Lara, P.Y. Joubert, and D. Placko. La caméra photothermique - partie ii : Applications industrielles, perspectives damélioration par un nouveau traitement dimage. *Instrumentation, Mesure, Métrologie*, 1(1):41–67, 2001.

[38] M. Lax. Temperature rise induced by a laser beam. *Journal of Applied Physics*, 48(9):3660, 1977.

[39] T. Li, D. P. Almond, and D.A. S. Rees. Crack imaging by scanning laser-line thermography and laser-spot thermography. *Measurement Science and Tecnhonolgy*, 22(3):035701, 2011.

[40] T. Li, D. P. Almond, and D.A. S. Rees. Crack imaging by scanning pulsed laser spot thermography. *NDT&E International*, 44(2):216–225, 2011.

[41] W. A. Link and R. J. Barker. *Bayesian Inference with ecological applications*. Academic Press, 2010.

[42] A Loeb. Gpu-heat-simulation. https://github.com/AndrewLoeb/GPU-heat-simulation/, October 2016.

[43] A. Loeb and C. Earls. Optimized inspection design for the thermographic characterization of sub-pixel sized through cracks. *NDT & E International*, 82:44–55, 2016.

[44] A. Logg, K.-A. Mardal, and G. N. Wells et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.

[45] D. D. Macdonald. Passivity–the key to our metals-based civilization. *Pure and Applied Chemistry*, 71(6):951–978, 1999.

[46] S. Marinetti and V. Vavilov. IR thermographic detection and characterization of hidden corrosion in metals: General analysis. *Corrosion Science*, 52(3):865–872, 2010.

[47] J. Martínez-Frutos and D. Herrero-Pérez. Efficient matrix-free gpu imple-

mentation of fixed grid finite element analysis. *Finite Elements in Analysis and Design*, 104:61–71, 2015.

[48] J. Martínez-Frutos and D. Herrero-Pérez. Large-scale robust topology optimization using multi-GPU systems. *Computer Methods in Applied Mechanics and Engineering*, 311:393–414, November 2016.

[49] J. Martínez-Frutos, P. J. Martínez-Castejón, and D. Herrero-Pérez. Fine-grained gpu implementation of assembly-free iterative solver for finite element problems. *Computers and Structures*, 157:9–18, 2015.

[50] Jesús Martínez-Frutos, Pedro J. Martínez-Castejón, and David Herrero-Pérez. Efficient topology optimization using GPU computing with multilevel granularity. *Advances in Engineering Software*, 106:47–62, 2017.

[51] E Müller, X Guo, R Scheichl, and S Shi. Matrix-free gpu implementation of a preconditioned conjugate gradient solver for anisotropic elliptic PDEs. *Computing and Visualization in Science*, 16(2):41–58, Apr 2013.

[52] Olivier Poisson. Uniqueness and hölder stability of discontinuous diffusion coefficients in three related inverse problems for the heat equation. *Inverse Problems*, 24(2), 2008.

[53] A. Rashed, D.P. Almond, D.A.S. Rees, S. Burrows, and S. Dixon. Crack detection by laser spot imaging thermography. pages 500–506. Review of Quantitative Nondestructive Evaluation, 2007.

[54] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[55] J. F. Ready. *Effects of High-Power Laser Radiation*. Academic Press, 2012.

[56] W. N. Reynolds. Thermographic methods applied to industrial materials. *Canadian Journal of Physics*, 64(9):1150–1154, 1986.

[57] P.R. Roberge and R.W. Revie. *Corrosion Inspection and Monitoring*. Wiley Series in Corrosion. Wiley, 2007.

[58] S.I. Rokhlin, J.-Y. Kim, H. Nagy, and B. Zoofan. Effect of pitting corrosion on fatigue crack initiation and fatigue life. *Engineering Fracture Mechanics*, 62(4-5):425–444, 1999.

[59] Mary P. Ryan, David E. Williams, Richard J. Chater, Bernie M. Hutton, and David S. McPhail. Why stainless steel corrodes. *Nature*, 415(6873):770–774, 2002.

[60] J. Schlichting, Ch. Maierhofer, and M. Kreutzbruck. Crack sizing by laser excited thermography. *NDT&E International*, 45(1):133–140, 2012.

[61] Stephan Schmidt and Volker Schulz. A 2589 line topology optimization code written for the graphics card. *Computing and Visualization in Science*, 14(6):249–256, 2011.

[62] S. Shepard. Introduction to active thermography for non-destructive evaluation. *Anti-Corrosion Methods and Materials*, 44(4):236–239, 1997.

[63] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Carnegie-Mellon University. Department of Computer Science, 1994.

[64] J. E. Stone, D. Gohara, and G. Shi. Opencl: A parallel programming standard for heterogeneous computing systems. *IEEE Design & Test*, 12(3):66–73, 2010.

[65] Daniel Tataru. Unique continuation problems for partial differential equations. In *Geometric Methods in Inverse Problems and PDE Control*, pages 239–255. Springer Nature, 2004.

[66] G. T.-N. Tsao. Thermal conductivity of two-phase materials. *Industrial & Engineering Chemistry*, 53(5):395–397, May 1961.

[67] J. Varis. *Detection of vertical cracks in carbon fiber composites using an infrared line scanner*, chapter 5, pages 1223–1227. Springer US, 1993.

[68] V. Vavilov, E. Grinzato, P.G. Bison, S. Marinetti, and M.J. Bales. Surface transient temperature inversion for hidden corrosion characterisation: theory and applications. *International Journal of Heat and Mass Transfer*, 39(2):355–371, 1996.

[69] V.P. Vavilov and D.D. Burleigh. Pulsed thermal ndt in tables, figures and formulas. pages 1–15. Thermosense: Thermal Infrared Applications XXXVII, 2015.

[70] Eddie Wadbro and Martin Berggren. Megapixel topology optimization on a graphics processing unit. *SIAM Review*, 51(4):707–721, November 2009.

[71] C. Zoppou and J.H. Knight. Analytical solution of a spatially variable coefficient advection–diffusion equation in up to three dimensions. *Applied Mathematical Modelling*, 23(9):667–685, Sep 1999.